4. (ORIGINAL) A method according to claim 3, wherein if at least one other object to which said object is related is classed as malware, then classifying said object as malware.

5. (ORIGINAL) A method according to claim 3, wherein said other objects include the object or similar objects stored on at least some of the remote computers.

6. (ORIGINAL) A method according to claim 3, wherein said other objects include other objects that are parent objects or child objects or otherwise process-related objects to said object.

7. (ORIGINAL) A method according to claim 1, wherein the data is sent in the form of key that is obtained by a hashing process carried out in respect of the objects on the respective remote computers.

8. (ORIGINAL) A method according to claim 7, wherein the key has at least one component that represents executable instructions contained within or constituted by the object.

9. (ORIGINAL) A method according to claim 7, wherein the key has at least one component that represents data about said object.

10. (ORIGINAL) A method according to claim 9, wherein said data about said object includes at least one of: the current name of the object; the physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object;

vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers.

11. (ORIGINAL) A method according to claim 7, wherein the key has at least one component that represents the physical size of the object.

12. (CANCELLED)

13. (CURRENTLY AMENDED) Apparatus for classifying a computer object as malware, the apparatus comprising:

a base computer constructed and arranged to receive data about a computer object from each of plural remote computers on which the object or similar objects are stored;

the base computer being constructed and arranged to ~~compare~~ determine whether the data about the computer object received from said plural computers indicates that the computer object is malware; and,

the base computer being constructed and arranged to classify the computer object as malware ~~on the basis of said comparison~~ when the base computer determines that the date indicates that the computer object is malware,

wherein the base computer is constructed and arranged to generate a mask for an object that is initially classed not as malware, said mask defining acceptable behaviour for the object, wherein ~~and the base computer or each of the plural remote computers is constructed and~~

~~arranged to monitor~~ operation of the object is automatically monitored when the object is running on at least one of the remote computers, the object being allowed to continue to run when behavior of the object is permitted by the mask, the object not being permitted to run when the actual monitored behaviour of the object extends beyond that permitted by the mask, and ~~to~~ ~~reclassify~~ the object being reclassified as malware [[if]] when the actual monitored behaviour extends beyond that permitted by the mask.

14. (CURRENTLY AMENDED) Apparatus according to claim 13, ~~wherein the base computer is constructed and arranged so as to compare the data where~~ the data includes one or more of: executable instructions contained within or constituted by the object; the size of the object; the current name of the object; the physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers.

15. (CURRENTLY AMENDED) Apparatus according to claim 13, wherein the base computer is constructed and arranged so that the ~~comparing~~ determining identifies relationships between the object and other objects.

16. (ORIGINAL) Apparatus according to claim 15, the base computer is constructed and arranged so that if at least one other object to which said object is related is classed as malware, then said object is classified as malware.

17. (ORIGINAL) Apparatus according to claim 15, wherein the base computer is constructed and arranged so that said other objects include the object or similar objects stored on at least some of said remote computers.

18. (ORIGINAL) Apparatus according to claim 15, wherein the base computer is constructed and arranged so that said other objects include other objects that are parent objects or child objects or otherwise process-related objects to said object.

19. (ORIGINAL) Apparatus according to claim 13, wherein the base computer is constructed and arranged so as to be able to process data that is sent in the form of key that is obtained by a hashing process carried out in respect of the objects on said respective remote computers.

20. (ORIGINAL) Apparatus according to claim 19, wherein the key has at least one component that represents executable instructions contained within or constituted by the object.

21. (ORIGINAL) Apparatus according to claim 19, wherein the key has at least one component that represents data about said object.

22. (ORIGINAL) Apparatus according to claim 21, wherein said data about said object includes at least one of: the current name of the object; the physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers.

23. (ORIGINAL) Apparatus according to claim 19, wherein the key has at least one component that represents the physical size of the object.

24. (CANCELLED)

25. (CURRENTLY AMENDED) A method of providing data about a computer object from a remote computer to a base computer so that a comparison can be made at the base computer with similar data received from other remote computers, the method comprising:

providing from a remote computer to a base computer data about a computer object that is stored on the remote computer;

the data including one or more of: executable instructions contained within or constituted by the object; the size of the object; the current name of the object; the physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the

object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers;

the data being sent in the form of key that is obtained by a hashing process carried out in respect of the object on the remote computer,

generating ~~receiving at the remote computer~~ a mask for the [[a]] computer object ~~stored at the remote computer~~, [[said]] the mask defining acceptable behaviour for the object;

executing the computer object at the remote computer; [[and]]

automatically monitoring operation of the computer object at the remote computer compared to the behaviour permitted by the mask;

allowing the object to continue to run when behavior of the computer object is permitted by the mask;

disallowing the object to run when the actual monitored behaviour of the object extends beyond that permitted by the mask; and

reclassifying the object as malware when the actual monitored behaviour extends beyond that permitted by the mask.

~~and, in the event that the behaviour of the object extends beyond that permitted by the mask, sending data to the base computer that the behaviour of the object has exceeded that permitted by the mask.~~

26. (ORIGINAL) A method according to claim 25, wherein the key has at least one component that represents executable instructions contained within or constituted by the object.

27. (ORIGINAL) A method according to claim 25, wherein the key has at least one component that represents data about said object.

28. (ORIGINAL) A method according to claim 27, wherein said data about said object includes at least one of: the current name of the object; the physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers.

29. (ORIGINAL) A method according to claim 25, wherein the key has at least one component that represents the physical size of the object.

30. (CURRENTLY AMENDED) A storage medium ~~carrier~~ storing a computer program comprising program instructions for causing a computer to perform the method of claim 25.

31. (WITHDRAWN) A method of determining the protection that a remote computer has from malware, the method comprising:

receiving at a base computer details of all or selected security products operating at a point in time on said remote computer;

receiving similar information from other remote computers connected to the base computer; and,

identifying any malware processes that were not identified by said other remote computers having that particular combination of security products.

32. (WITHDRAWN) A method according to claim 31, comprising providing information to the user of said remote computer that said remote computer may be susceptible to attack by said malware processes on the basis of said identifying.

33. (WITHDRAWN) A method according to claim 31, wherein the details of all or selected security products includes the name of the security products, versions, and loaded signature files.

34. (WITHDRAWN) Apparatus for determining the protection that a remote computer has from malware, the apparatus comprising:

a base computer constructed and arranged to receive computer details of all or selected security products operating at a point in time on said remote computer;

the base computer being constructed and arranged to receive similar information from other remote computers connected to the base computer; and,

the base computer being constructed and arranged to identify any malware processes that were not identified by said other remote computers having that particular combination of security products.

35. (WITHDRAWN) Apparatus according to claim 34, wherein the base computer is constructed and arranged to provide information to the user of said remote computer that said

remote computer may be susceptible to attack by said malware processes on the basis of said identifying.

36. (WITHDRAWN) Apparatus according to claim 34, wherein the details of all or selected security products includes the name of the security products, versions, and loaded signature files.

37. (WITHDRAWN) A method of classifying a computer object as malware or as safe, wherein said computer object is a descendant or otherwise related object of a first computer object, the method comprising:

classifying a first computer object as malware or as safe;

identifying in a key relating to said first computer object a component that uniquely identifies the first computer object and that is inherited or otherwise present in the key of a descendant or other related computer object of the first computer object; and,

classifying said computer object as malware or as safe as the case may be on the basis of the unique identifier component being present in the key of said computer object.

38. (WITHDRAWN) A method according to claim 37, comprising monitoring activities of said first computer object and reclassifying the first computer object as malware in the case that it was initially classified as safe and subsequently determined to be malware, the method further comprising automatically classifying as malware any computer object that has a key in which said unique identifier component is present.

39. (WITHDRAWN) A computer program comprising program instructions for causing a computer to perform the method of claim 38.

40. (WITHDRAWN) Apparatus for classifying a computer object as malware or as safe, wherein said computer object is a descendant or otherwise related object of a first computer object, the apparatus comprising:

a computer constructed and arranged to classify a first computer object as malware or as safe;

the computer being constructed and arranged to identify in a key relating to said first computer object a component that uniquely identifies the first computer object and that is inherited or otherwise present in the key of a descendant or other related computer object of the first computer object; and,

the computer being constructed and arranged to classify said computer object as malware or as safe as the case may be on the basis of the unique identifier component being present in the key of said computer object.

41. (WITHDRAWN) Apparatus according to claim 40, wherein the computer is constructed and arranged to monitor activities of said first computer object and reclassify the first computer object as malware in the case that it was initially classified as safe and subsequently determined to be malware, the computer further being constructed and arranged to automatically classify as malware any computer object that has a key in which said unique identifier component is present.

42. (WITHDRAWN) A method of installing software on a computer, the method comprising:

on initiation of installation of software on a computer, providing a computer-generated prompt on the computer to a user to ascertain whether the user authorises the installation; and,

ceasing the installation if a user authorisation is not received, else:

receiving at the computer the user's authorisation to proceed with the installation;

proceeding with the installation;

obtaining data about computer objects that are created or used during the installation;

storing said data at the local computer.

43. (WITHDRAWN) A method according to claim 42, wherein the locally stored data is referred to during the installation to ensure that all objects created or used during the installation are part of the installation process, and, if it is found that objects created or used during the installation are not part of the installation process, either or both of: (i) ceasing the installation and (ii) providing a computer-generated prompt on the computer to the user accordingly.

44. (WITHDRAWN) A method according to claim 42, comprising connecting the computer to a community database that is connectable to a plurality of computers, and uploading the stored data to the community database for comparison with similar data provided by other computers.

45. (WITHDRAWN) A method according to claim 42, comprising downloading data about trusted installers to the computer, said data about trusted installers being referred to during the

installation such that any objects relating to or created by the trusted installer are automatically authorised to proceed.

46. (WITHDRAWN) A method according to claim 45, wherein said data about trusted installers is referred to only for a predetermined time period following receipt at the computer of the user's authorisation to proceed with the installation.

47. (WITHDRAWN) A computer program comprising program instructions for causing a computer to perform the method of claim 42.

48. (NEW) A method according to claim 1, further comprising:

receiving information at the base computer about the behaviour of the object running on at least one of the remote computers;

wherein the generating of the mask for the object comprises building the mask with the base computer in accordance with normal behaviour of the object determined from the received information.

49. (NEW) A method according to claim 1, wherein:

if the monitored operation of the object running on the remote computer exhibits non-permitted behaviour that extends beyond that permitted by the mask, comparing at the base computer the non-permitted behaviour with the behaviour of at least one other object;

if the comparison indicates that the non-permitted behaviour is known to be not malicious for the at least one other object, allowing the non-permitted behaviour for the object; and

if the comparison indicates that the non-permitted behaviour is known to be not malicious for the at least one other object, modifying the mask to allow the non-permitted behaviour for the object in the future.

50. (NEW) An apparatus according to claim 13, wherein:

the base computer is constructed and arranged to receive information about the behaviour of the object running on at least one of the remote computers; and

the base computer is constructed and arranged to generate the mask in accordance with normal behaviour of the object determined from the received information.

51. (NEW) An apparatus according to claim 13, wherein:

the base computer is constructed and arranged to compare the non-permitted behaviour with the behaviour of at least one other object if the monitored operation of the object running on the remote computer exhibits non-permitted behaviour that extends beyond that permitted by the mask;

the base computer is constructed and arranged to allow the non-permitted behaviour for the object if the comparison indicates that the non-permitted behaviour is known to be not malicious for the at least one other object; and

the base computer is constructed and arranged to modify the mask to allow the non-permitted behaviour for the object in the future if the non-permitted behaviour is known to be not malicious for the at least one other object.

Interview Summary	Application No.	Applicant(s)	
	11/477,807	MORRIS ET AL.	
	Examiner	Art Unit	
	Cordelia Zecher	2432	

All participants (applicant, applicant's representative, PTO personnel):

(1) *Cordelia Zecher*.

(2) *Dale Lazar*.

(3) *Nicholas Panno*.

(4) _____.

Date of Interview: *12 January 2011*.

Type: a)☐ Telephonic b)☐ Video Conference
c)☒ Personal [copy given to: 1)☐ applicant 2)☒ applicant's representative]

Exhibit shown or demonstration conducted: d)☐ Yes e)☒ No.
If Yes, brief description: _____.

Claim(s) discussed: *1,13 and 25*.

Identification of prior art discussed: *Dozortsev*.

Agreement with respect to the claims f)☐ was reached. g)☒ was not reached. h)☐ N/A.

Substance of Interview including description of the general nature of what was agreed to if an agreement was reached, or any other comments: *Possible claim amendments were discussed to clarify the mask and allowing the object to run when the behaviour is permitted by the mask. Such amendments would require further search and consideration*.

(A fuller description, if necessary, and a copy of the amendments which the examiner agreed would render the claims allowable, if available, must be attached. Also, where no copy of the amendments that would render the claims allowable is available, a summary thereof must be attached.)

THE FORMAL WRITTEN REPLY TO THE LAST OFFICE ACTION MUST INCLUDE THE SUBSTANCE OF THE INTERVIEW. (See MPEP Section 713.04). If a reply to the last Office action has already been filed, APPLICANT IS GIVEN A NON-EXTENDABLE PERIOD OF THE LONGER OF ONE MONTH OR THIRTY DAYS FROM THIS INTERVIEW DATE, OR THE MAILING DATE OF THIS INTERVIEW SUMMARY FORM, WHICHEVER IS LATER, TO FILE A STATEMENT OF THE SUBSTANCE OF THE INTERVIEW. See Summary of Record of Interview requirements on reverse side or on attached sheet.

Trend Micro, Inc.
EX1007-448

Summary of Record of Interview Requirements

Manual of Patent Examining Procedure (MPEP), Section 713.04, Substance of Interview Must be Made of Record
A complete written statement as to the substance of any face-to-face, video conference, or telephone interview with regard to an application must be made of record in the application whether or not an agreement with the examiner was reached at the interview.

Title 37 Code of Federal Regulations (CFR) § 1.133 Interviews
Paragraph (b)

In every instance where reconsideration is requested in view of an interview with an examiner, a complete written statement of the reasons presented at the interview as warranting favorable action must be filed by the applicant. An interview does not remove the necessity for reply to Office action as specified in §§ 1.111, 1.135. (35 U.S.C. 132)

37 CFR §1.2 Business to be transacted in writing.
All business with the Patent or Trademark Office should be transacted in writing. The personal attendance of applicants or their attorneys or agents at the Patent and Trademark Office is unnecessary. The action of the Patent and Trademark Office will be based exclusively on the written record in the Office. No attention will be paid to any alleged oral promise, stipulation, or understanding in relation to which there is disagreement or doubt.

The action of the Patent and Trademark Office cannot be based exclusively on the written record in the Office if that record is itself incomplete through the failure to record the substance of interviews.

It is the responsibility of the applicant or the attorney or agent to make the substance of an interview of record in the application file, unless the examiner indicates he or she will do so. It is the examiner's responsibility to see that such a record is made and to correct material inaccuracies which bear directly on the question of patentability.

Examiners must complete an Interview Summary Form for each interview held where a matter of substance has been discussed during the interview by checking the appropriate boxes and filling in the blanks. Discussions regarding only procedural matters, directed solely to restriction requirements for which interview recordation is otherwise provided for in Section 812.01 of the Manual of Patent Examining Procedure, or pointing out typographical errors or unreadable script in Office actions or the like, are excluded from the interview recordation procedures below. Where the substance of an interview is completely recorded in an Examiners Amendment, no separate Interview Summary Record is required.

The Interview Summary Form shall be given an appropriate Paper No., placed in the right hand portion of the file, and listed on the "Contents" section of the file wrapper. In a personal interview, a duplicate of the Form is given to the applicant (or attorney or agent) at the conclusion of the interview. In the case of a telephone or video-conference interview, the copy is mailed to the applicant's correspondence address either with or prior to the next official communication. If additional correspondence from the examiner is not likely before an allowance or if other circumstances dictate, the Form should be mailed promptly after the interview rather than with the next official communication.

The Form provides for recordation of the following information:
- Application Number (Series Code and Serial Number)
- Name of applicant
- Name of examiner
- Date of interview
- Type of interview (telephonic, video-conference, or personal)
- Name of participant(s) (applicant, attorney or agent, examiner, other PTO personnel, etc.)
- An indication whether or not an exhibit was shown or a demonstration conducted
- An identification of the specific prior art discussed
- An indication whether an agreement was reached and if so, a description of the general nature of the agreement (may be by attachment of a copy of amendments or claims agreed as being allowable). Note: Agreement as to allowability is tentative and does not restrict further action by the examiner to the contrary.
- The signature of the examiner who conducted the interview (if Form is not an attachment to a signed Office action)

It is desirable that the examiner orally remind the applicant of his or her obligation to record the substance of the interview of each case. It should be noted, however, that the Interview Summary Form will not normally be considered a complete and proper recordation of the interview unless it includes, or is supplemented by the applicant or the examiner to include, all of the applicable items required below concerning the substance of the interview.

A complete and proper recordation of the substance of any interview should include at least the following applicable items:

1) A brief description of the nature of any exhibit shown or any demonstration conducted,
2) an identification of the claims discussed,
3) an identification of the specific prior art discussed,
4) an identification of the principal proposed amendments of a substantive nature discussed, unless these are already described on the Interview Summary Form completed by the Examiner,
5) a brief identification of the general thrust of the principal arguments presented to the examiner,
   (The identification of arguments need not be lengthy or elaborate. A verbatim or highly detailed description of the arguments is not required. The identification of the arguments is sufficient if the general nature or thrust of the principal arguments made to the examiner can be understood in the context of the application file. Of course, the applicant may desire to emphasize and fully describe those arguments which he or she feels were or might be persuasive to the examiner.)
6) a general indication of any other pertinent matters discussed, and
7) if appropriate, the general results or outcome of the interview unless already described in the Interview Summary Form completed by the examiner.

Examiners are expected to carefully review the applicant's record of the substance of an interview. If the record is not complete and accurate, the examiner will give the applicant an extendable one month time period to correct the record.

Examiner to Check for Accuracy

If the claims are allowable for other reasons of record, the examiner should send a letter setting forth the examiner's version of the statement attributed to him or her. If the record is complete and accurate, the examiner should place the indication, "Interview Record OK" on the paper recording the substance of the interview along with the date and the examiner's initials.

Interview Summary	Application No.	Applicant(s)
	11/477,807	MORRIS ET AL.
	Examiner	Art Unit
	Cordelia Zecher	2432

All participants (applicant, applicant's representative, PTO personnel):

(1) *Cordelia Zecher*.

(2) *Dale Lazar*.

(3) *Nicholas Panno*.

(4)____.

Date of Interview: *12 January 2011*.

Type: a)☐ Telephonic b)☐ Video Conference
c)☒ Personal [copy given to: 1)☐ applicant 2)☒ applicant's representative]

Exhibit shown or demonstration conducted: d)☐ Yes e)☒ No.
If Yes, brief description: _____.

Claim(s) discussed: *1,13 and 25*.

Identification of prior art discussed: *Dozortsev*.

Agreement with respect to the claims f)☐ was reached. g)☒ was not reached. h)☐ N/A.

Substance of Interview including description of the general nature of what was agreed to if an agreement was reached, or any other comments: *Possible claim amendments were discussed to clarify the mask and allowing the object to run when the behaviour is permitted by the mask. Such amendments would require further search and consideration*.

(A fuller description, if necessary, and a copy of the amendments which the examiner agreed would render the claims allowable, if available, must be attached. Also, where no copy of the amendments that would render the claims allowable is available, a summary thereof must be attached.)

THE FORMAL WRITTEN REPLY TO THE LAST OFFICE ACTION MUST INCLUDE THE SUBSTANCE OF THE INTERVIEW. (See MPEP Section 713.04). If a reply to the last Office action has already been filed, APPLICANT IS GIVEN A NON-EXTENDABLE PERIOD OF THE LONGER OF ONE MONTH OR THIRTY DAYS FROM THIS INTERVIEW DATE, OR THE MAILING DATE OF THIS INTERVIEW SUMMARY FORM, WHICHEVER IS LATER, TO FILE A STATEMENT OF THE SUBSTANCE OF THE INTERVIEW. See Summary of Record of Interview requirements on reverse side or on attached sheet.

Trend Micro, Inc.
EX1007-450

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/477,807	06/30/2006	Melvyn Morris	339000-000003	4785

47604 7590 09/28/2010

DLA PIPER LLP US
P. O. BOX 2758
RESTON, VA 20195

EXAMINER	
ZECHER, CORDELIA P K	
ART UNIT	PAPER NUMBER
2432	
MAIL DATE	DELIVERY MODE
09/28/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	11/477,807	MORRIS ET AL.	
	Examiner	Art Unit	
	Cordelia Zecher	2432	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *07 September 2010*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-11,13-23 and 25-30* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-11,13-23 and 25-30* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

Trend Micro, Inc.
EX1007-452

DETAILED ACTION

Response to Arguments

1. Applicant's arguments filed September 7, 2010 have been fully considered but they are not persuasive. Applicant argues that amending claim 30 to recite a carrier storing a computer program places the claim in one of the four statutory categories of invention. However, in paragraph [0071] of applicant's specification, applicant defined carrier to include signals. Signals do not fall into one of the four statutory categories of invention and are therefore non-statutory.

2. Applicant argues that Dozortsev in view of Sheymov fails to teach or suggest comparing data about objects received from plural computers across the community to determine whether or not an object is malware. However, Dozortsev teaches comparing the executable code signature with entries in the local database (page 5, paragraph 44) and then forwarding the signature on to the central computer to compare with the central database to determine if the code is flagged as malware (page 5, paragraph 45). Dozortsev does not explicitly disclose plural remote computers. However, Sheymov teaches having both centralized and distributed databases (page 2, paragraph 23). Therefore, in combination, Dozortsev in view of Sheymov teaches the limitation of the claim.

3. Applicant argues that Dozortsev in view of Sheymov fails to teach or suggest monitoring execution of an object against a mask of acceptable behavior for that object and reclassifying an object as malware at the central computer if the object exceeds that behavior. However, Dozortsev teaches when code is not known malware, that the

code may continue to execute, and therefore not be considered malware (page 4, paragraphs 37-38), then the code is investigated and executed in an environment to see if it meets the acceptable behavior, and when the code behaves as malware, classifying the code as malware (page 4, paragraphs 40-41). Therefore Dozortsev teaches the limitation of the claim.

Specification

The following guidelines illustrate the preferred layout for the specification of a utility application. These guidelines are suggested for the applicant's use.

Arrangement of the Specification

As provided in 37 CFR 1.77(b), the specification of a utility application should include the following sections in order. Each of the lettered items should appear in upper case, without underlining or bold type, as a section heading. If no text follows the section heading, the phrase "Not Applicable" should follow the section heading:

(a) TITLE OF THE INVENTION.
(b) CROSS-REFERENCE TO RELATED APPLICATIONS.
(c) STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT.
(d) THE NAMES OF THE PARTIES TO A JOINT RESEARCH AGREEMENT.
(e) INCORPORATION-BY-REFERENCE OF MATERIAL SUBMITTED ON A COMPACT DISC.
(f) BACKGROUND OF THE INVENTION.
    (1) Field of the Invention.
    (2) Description of Related Art including information disclosed under 37 CFR 1.97 and 1.98.
(g) BRIEF SUMMARY OF THE INVENTION.
(h) BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWING(S).
(i) DETAILED DESCRIPTION OF THE INVENTION.
(j) CLAIM OR CLAIMS (commencing on a separate sheet).
(k) ABSTRACT OF THE DISCLOSURE (commencing on a separate sheet).

(l) SEQUENCE LISTING (See MPEP § 2424 and 37 CFR 1.821-1.825. A "Sequence Listing" is required on paper if the application discloses a nucleotide or amino acid sequence as defined in 37 CFR 1.821(a) and if the required "Sequence Listing" is not submitted as an electronic document on compact disc).

4. The disclosure is objected to because of the following informalities: The specification is not divided into the appropriate sections as listed above.

5. The lengthy specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

6. Appropriate correction is required.

7. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

Claim Rejections - 35 USC § 101

8. Claim 30 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The claim is directed to a carrier. In the specification, applicant defines a carrier to include a signal ([0071]). *See* MPEP 2111.01. When the broadest reasonable interpretation of a claim covers a signal *per se*, the claim must be rejected under 35 U.S.C. §101 as covering non-statutory subject matter. *See In re Nuijten*, 500 F.3d 1346, 1356-57 (Fed. Cir. 2007) (transitory embodiments are not direct to statutory subject matter) and *Interim Examination*

Instructions for Evaluating Subject Matter Eligibility Under 35 U.S.C §101, Aug. 24, 2009; p. 2.

Claim Rejections - 35 USC § 103

9. Claims 1 – 9, 12 – 21, 24 – 27, and 30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dozortsev's US Publication 2003/0177394 A1 in view of Sheymov et al's US Publication 2001/0052014 A1.

10. Referring to claims 1 and 13, Dozortsev teaches:

a. At a base computer, receiving data about a computer object from a remote computer on which the object or similar objects are stored (page 5, paragraph 44).

b. Comparing in the base computer the data about the computer object received from the plural computers (page 5, paragraph 44).

c. Classifying the computer object as malware on the basis of said comparison (page 5, paragraph 44).

d. When the computer object is not known malware, initially classifying an object as not malware (page 4, paragraphs 37-38).

e. Generating a mask for said object that defines acceptable behavior for the object (page 4, paragraph 40).

f. Monitoring operation of said object and reclassifying the object as malware if the actual monitored behavior extends beyond that permitted by the mask (page 4, paragraphs 40-41).

2. Dozortsev does not explicitly teach receiving data from plural remote computers. However, Sheymov teaches having both centralized and distributed databases of information so there is a greater chance of being able to react to the attempt in real time (page 2, paragraph 23). Dozortsev and Sheymov are from the same field of endeavor, network protection. At the time of invention, it would have been obvious to modify the database of Dozortsev to include the multiple databases of Sheymov. The suggestion for doing so would have been to have a greater chance to react to the attack in real time (page 2, paragraph 23).

3. Referring to claims 2 and 14, Dozortsev teaches that the data about the computer object that is sent from the remote computer to the base computer and that is used in the comparison includes executable instructions contained within or constituted by the subject (page 3, paragraph 21).

4. Referring to claims 3 and 15, Dozortsev teaches comparing identifies relationships between the object and other objects (page 3, paragraph 22).

5. Referring to claims 4 and 16, Dozortsev teaches wherein if at least one other object to which said object is related is classified as malware, then classifying said object as malware (page 3, paragraph 25).

6. Referring to claims 5 and 17, Dozortsev teaches the other objects include objects stored on at least some of the remote computers (page 5, paragraphs 43-44).

7. Referring to claims 6 and 18, Dozortsev teaches that the other objects include otherwise process related objects (page 5, paragraph 44).

8. Referring to claims 7 and 19, Dozortsev teaches that the data is sent in the form of a key that is obtained by a hashing process carried out in respect of the objects on the remote computer (page 3, paragraph 21).

9. Referring to claims 8 and 20, Dozortsev teaches that the key has one component that represents executable instructions contained within or constituted by the object (page 3, paragraph 21).

10. Referring to claims 9 and 21, Dozortsev teaches that the key has at least one component that represents data about said object (page 3, paragraph 21).

11. Referring to claims 25 and 30, Dozortsev teaches:

g. Providing from a remote computer to a base computer data about a computer object that is stored on the remote computer (page 5, paragraph 44).

h. The data including executable instructions contained within or constituted by the object (page 3, paragraph 21).

i. The data being sent in the form of key that is obtained by a hashing process carried out in respect of the object on the remote computer (page 3, paragraph 21).

j. Receiving at the remote computer a mask for a computer object stored at the remote computer, said mask defining acceptable behavior of the object (page 4, paragraph 40).

k. Executing the computer object at the remote computer (page 4, paragraph 40).

l. Monitoring operation of the computer object at the remote computer compared to the behavior permitted by the mask, and in the event that the behavior of the object extends beyond that permitted by the mask, sending data to the base computer that the behavior of the object has exceeded that permitted by the mask (page 4, paragraphs 40-41).

12. Dozortsev does not explicitly teach receiving data from plural remote computers. However, Sheymov teaches having both centralized and distributed databases of information so there is a greater chance of being able to react to the attempt in real time (page 2, paragraph 23). Dozortsev and Sheymov are from the same field of endeavor, network protection. At the time of invention, it would have been obvious to modify the database of Dozortsev to include the multiple databases of Sheymov. The suggestion for doing so would have been to have a greater chance to react to the attack in real time (page 2, paragraph 23).

13. Referring to claim 26, Dozortsev teaches that the key has one component that represents executable instructions contained within or constituted by the object (page 3, paragraph 21).

14. Referring to claim 27, Dozortsev teaches that the key has at least one component that represents data about said object (page 3, paragraph 21).

15. Claims 10, 11, 22, 23, 28, and 29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dozortsev in view of Sheymov as applied above, and further in view of Barton et al's US Publication 2005/0021994 A1. Dozortsev in view of Sheymov

discloses all the limitations of the parent claims. Dozortsev in view of Sheymov does not explicitly disclose the data about the object including the name of the object, the location or the size of the object. However, Barton discloses indicating for the object the file name, size and location (pages 2-3, paragraph 39). Dozortsev in view of Sheymov and Barton are analogous art because they are from the same field of endeavor, malware detection. At the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teachings of Dozortsev in view of Sheymov and Barton before him or her, to modify the signature of Dozortsev in view of Sheymov to include the data name and size of Barton. The suggestion/motivation for doing so would have been the characteristics are strongly indicative of a particular file being one of the candidates (page 1, paragraph 14).

Conclusion

16. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Cordelia Zecher whose telephone number is (571)272-7771. The examiner can normally be reached on Monday - Thursday 8:00 - 5:00 EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Gilberto Barron can be reached on 571-272-3799. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/C. Z./
Examiner, Art Unit 2432

/Benjamin E Lanier/
Primary Examiner, Art Unit 2432

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11477807	MORRIS ET AL.
	Examiner	Art Unit
	Cordelia Zecher	2432

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	04/28/2010	09/23/2010							
	1	✓	✓							
	2	✓	✓							
	3	✓	✓							
	4	✓	✓							
	5	✓	✓							
	6	✓	✓							
	7	✓	✓							
	8	✓	✓							
	9	✓	✓							
	10	✓	✓							
	11	✓	✓							
	12	✓	-							
	13	✓	✓							
	14	✓	✓							
	15	✓	✓							
	16	✓	✓							
	17	✓	✓							
	18	✓	✓							
	19	✓	✓							
	20	✓	✓							
	21	✓	✓							
	22	✓	✓							
	23	✓	✓							
	24	✓	-							
	25	✓	✓							
	26	✓	✓							
	27	✓	✓							
	28	✓	✓							
	29	✓	✓							
	30	✓	✓							
	31	N								
	32	N								
	33	N								
	34	N								
	35	N								
	36	N								

Trend Micro, Inc.
EX1007-462

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11477807	MORRIS ET AL.
	Examiner	**Art Unit**
	Cordelia Zecher	2432

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	04/28/2010	09/23/2010							
	37	N								
	38	N								
	39	N								
	40	N								
	41	N								
	42	N								
	43	N								
	44	N								
	45	N								
	46	N								
	47	N								

Trend Micro, Inc.
EX1007-463

DOCKET NO. 339000-000003 PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of: MORRIS, Melvyn et al. Confirmation No. 4785

Application No. 11/477,807 Group Art Unit: 2432

Filed: June 30, 2006 Examiner: C. Zecher

For: METHODS AND APPARATUS FOR DEALING WITH MALWARE

AMENDMENT UNDER 37 C.F.R. 1.111

Assistant Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

September 7, 2010

Dear Sir:

This is in response to the Office Action dated May 4, 2010, and having a period for response set to expire on August 4, 2010. A petition for a one-month extension of time, together with the requisite fee for same, is submitted herewith, thereby extending the period for response to September 4, 2010.

The following amendments and remarks are respectfully submitted. Reconsideration of the claims is respectfully requested.

Amendments to the Claims begin on page 2 of this paper.

Remarks/Arguments begin on page 15 of this paper.

Trend Micro, Inc.
EX1007-464

Remarks/Arguments

The application has been amended to place the application condition for allowance. All amendments are supported by the specification. An early notice of allowance is earnestly solicited.

Claims 1 through 11, 13 through 23 and 25 through 30 remain pending. Claims 12 and 24 have been deleted. Claims 31 through 47 were previously withdrawn. Claim 1, 13, 25 and 30 have been amended.

The disclosure stands objected to for not being divided into appropriate sections. Applicants traverse this objection.

The Office Action correctly characterizes the arrangement specified in 37 C.F.R. 1.77 (b) as "guidelines" of a "preferred" layout that is "suggested" for the applicant's use. Thus, the Office Action itself acknowledges that the arrangement is discretionary on the part of the applicant. In this case, applicant declines the suggestion in the Office Action.

Claim 30 stands rejected under 35 U.S.C. §101 as being directed to non-statutory subject matter. Claim 30 has been amended to be directed to a manufacture, one of the four statutory categories of invention.

Claims 1-9, 12-21, 24-27 and 30 stand rejected as obvious over Dozortsev and Sheymov. The claims have been amended to obviate this rejection.

Trend Micro, Inc.
EX1007-465

As mentioned in the introduction to the present application, the conventional approach to identifying malware is to maintain a database of signatures identifying known bad objects, i.e. malware. The database may be stored locally on the remote computer and/or at a central base computer/server. The signature is conventionally made by creating a hash or checksum of the object file, which uniquely identifies that object. The signatures of objects at remote computers are compared to the database to look up whether they match known malware. If the signature of the object is not found the database, they are assumed safe or else referred for further investigation by a human analyst if the object is thought to be suspicious. The drawback of this approach is that considerable effort must be undertaken by providers of such systems in identifying and analysing new malware and generating signatures of objects that are found to be bad. As the volume of malware increases, it is proving practically an impossible task to keep up with identifying suspect objects, and investigating whether or not they are bad. For example, large vendors of anti-malware packages using this system typically employ many thousands of human analysts working on identifying and analysing objects to keep the database of signatures of bad objects reasonably up to date.

The present invention adopts a very different and inventive approach to protecting against malware. The approach is founded on receiving information about objects present on a plurality of remote computers at a base computer. This information about objects in the community, i.e. from plural remote computers, is aggregated at the base computer. This allows analysis to be performed mapping the behaviour and attributes of objects known across the community in order to identify suspicious behaviour and to identify malware at an early stage. This approach allows among, other advantages, the number of human analysts needed to be massively reduced. This

Trend Micro, Inc.
EX1007-466

can also reduce the latency with which new threats are identified and responding to objects exhibiting new, potentially malevolent behaviour.

Dozortsev teaches an approach using the standard technique of maintaining a database of signatures of known bad objects, i.e. malware, and comparing new objects with the database of known bad objects to see if they are malware. Dozortsev differs slightly from the normal approach of downloading the database of bad objects to the remote computer, by having a central database of bad objects. Remote computers send signatures of new objects to the base computer to be compared centrally with the database. However, this approach still fundamentally relies on using a database of signatures of bad objects. The signatures in the database must be generated by having human analysts investigating suspected bad objects. For example, as discussed at paragraph 47 and 49 of Dozortsev, the objects are analysed centrally by a human analyst by running the file in a "sandbox" environment or by examining the code or other such slow and laborious techniques. This still suffers from the problems of spotting suspicious objects in the first place, and in the huge effort required to analyse suspicious objects to keep the database of threats up to date.

In contrast, the present invention requires receiving data about a computer object from each of plural remote computers on which the object or similar objects are stored; comparing in the base computer the data about the computer object received from the plural computers; and, classifying the computer object as malware on the basis of said comparison. In effect, the invention builds up a central picture of objects and their interrelationships and activities across the entire community. This allows automation of the process of identifying malware by aggregating and comparing activity of objects running across the community, i.e. on plural

Trend Micro, Inc.
EX1007-467

remote computers. Many examples are given in the present description about the sophisticated analysis that may be performed on this data.

Furthermore, amended claim 1 requires when an object is not known malware, initially classifying the object as not malware, generating a mask for said object that defines acceptable behaviour for the object, and comprising monitoring operation of the object on at least one of the remote computers and reclassifying the object as malware if the actual monitored behaviour extends beyond that permitted by the mask. This has advantages in dealing with objects that do not initially exhibit suspicious behaviour, but later start to exhibit malevolent behaviour. In the prior art, such as Dozortsev, an object can only be marked as good or bad, i.e. a binary decision. This does not permit the system to automatically deal with the case where an object only starts to exhibits malevolent behaviour in the future, which is a common attribute of certain types of malware. The present invention deals with this situation by initially classifying the object as good but also generating an appropriate behaviour mask for the object which is used to monitor the permitted behaviour of the object. If the object exceeds the permitted behaviour, then this information can be sent to the base computer, which can take appropriate action, i.e. reclassifying the object as bad. Thus, the invention has the capacity of changing the classification of an object if malevolent behaviour emerges later on.

As an example, suppose a new version of Internet Explorer appeared. This could be a legitimate update to Internet Explorer released by Microsoft or alternatively it could be a file infected with a virus. In the prior art, the new object would have an unknown signature, so an in-house analyst would laboriously analyse the new object and determine whether or not it was safe. Whilst this analysis is carried out, the object would either be blocked, which would cause huge

inconvenience to users of the new object, or allowed to run, in which case there is a risk of the object performing malevolent acts. In contrast, the present invention would collect data at the base computer from remote computers running the new version of Internet Explorer. Using the information collected, the system could determine that the new object purports to be a new version of Internet Explorer. However, it may not be apparent at this point whether or not the new object is capable of malevolent behaviour. In this scenario the present invention generates an appropriate behavioural mask for the object, e.g. by using a profile of behaviour of previous versions of Internet Explorer that are known not to be malware, or by using a profile for the behaviour appropriate for a web browser. The remote computers are allowed to let the new version run whilst monitoring its behaviour against the mask. The instant the new object exhibits some new, malevolent behaviour, this can be stopped at the remote computer, as well as being flagged to the base computer and used at the base computer to change the classification of the object. Thus, the present invention allows an instant response to an object changing its behaviour to exhibit malevolent behaviour in the future.

Dozortsev does not teach or suggest anything like this scheme. The examiner referred to paragraphs 37 to 40 of Dozortsev as being relevant to the behavioural mask feature of the present invention (in previous claim 12). However, these paragraphs teach only that unknown objects received at the base computer may be allowed to run whilst being investigated by a human analyst (paragraph 39). There is no mention in Dozortsev of using a behavioural mask. Rather, in Dozortsev the investigation into the new object can take the form of allowing the program to run in a sandbox environment and/or analysing the code of the object (see paragraph 41). However, this is merely the standard technique of investigating the new object by in-house

analysts which is becoming increasingly inadequate to cope with the volume of malware and the speed at which new malware is introduced and exhibits new behaviour.

Thus, in short, Dozortsev does not teach or suggest the concepts of (i) comparing data about objects received from plural computers across the community to determine whether or not an object is malware, and (ii) monitoring the execution of an object against a mask of acceptable behaviour for that object and reclassifying an object as malware at the central computer if the object exceeds that behaviour.

Sheymov does not compensate for the deficiencies of Dozortsev. None of the other prior art cited by the examiner teaches or suggests these concepts either. Thus, the present invention as defined by independent claims 1, 13 and 25 is novel and non-obvious over these documents.

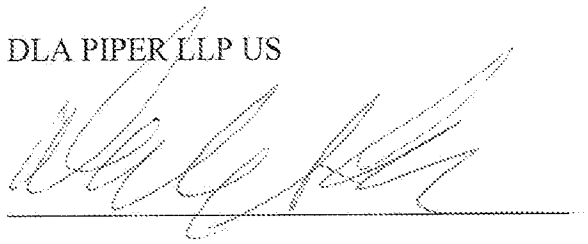
Claims 10, 11, 22, 23, 28 and 29 stand rejected as obvious over Dozortsev, Sheymov and Barton. Barton is cited for disclosing the step of indicating the file name, size and location of an object. Barton fails to compensate for the deficiencies of Dozortsev and Sheymon noted above relative to the independent claims. Since claims 10, 11, 22, 23, 28 and 29 are dependent on the independent claims, they are patentable for the reasons discussed above for claim 1.

**Trend Micro, Inc.
EX1007-470**

Please charge any shortage in the fees or credit any overpayment to Deposit Account No. 50-3266.

Respectfully submitted,

DLA PIPER LLP US



Dale S. Lazar
Registration No. 28,872
Attorney for Applicant

DSL/jcc
PO Box 2758
Reston, VA 20195
(703) 773.4000 Telephone
(703) 773.5200 Fax

Trend Micro, Inc.
EX1007-471

Electronic Patent Application Fee Transmittal

Application Number:	11477807
Filing Date:	30-Jun-2006
Title of Invention:	Methods and apparatus for dealing with malware
First Named Inventor/Applicant Name:	Melvyn Morris
Filer:	Dale S. Lazar/Josefina Cheng
Attorney Docket Number:	339000-000003

Filed as Large Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Extension - 1 month with $0 paid	1251	1	130	130

Trend Micro, Inc.
EX1007-472

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
			Total in USD ($)	**130**

Electronic Acknowledgement Receipt

EFS ID:	8368847
Application Number:	11477807
International Application Number:	
Confirmation Number:	4785
Title of Invention:	Methods and apparatus for dealing with malware
First Named Inventor/Applicant Name:	Melvyn Morris
Customer Number:	47604
Filer:	Dale S. Lazar/Josefina Cheng
Filer Authorized By:	Dale S. Lazar
Attorney Docket Number:	339000-000003
Receipt Date:	07-SEP-2010
Filing Date:	30-JUN-2006
Time Stamp:	18:42:30
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Credit Card
Payment was successfully received in RAM	$130
RAM confirmation Number	5133
Deposit Account	
Authorized User	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)

Trend Micro, Inc.
EX1007-474

1	Extension of Time	339000-3EOT.pdf	483680 58a7dd941d5de4bed16f855d8a3dfc9abe70e6c2	no	1
Warnings:					
Information:					
2		339000-3Amendment.pdf	6679379 208823a06f186a3d269975407f0d3ee7e21c8cf5	yes	21

	Multipart Description/PDF files in .zip description		
	Document Description	**Start**	**End**
	Amendment/Req. Reconsideration-After Non-Final Reject	1	1
	Claims	2	14
	Applicant Arguments/Remarks Made in an Amendment	15	21

Warnings:					
Information:					
3	Fee Worksheet (PTO-875)	fee-info.pdf	29850 42a5e64b1dc661c83b04acb45983a7d7ed42726b	no	2
Warnings:					
Information:					
			Total Files Size (in bytes):	7192909	

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PTO/SB/22 (10-04)
Approved for use through 7/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE
Under the paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless if displays a valid OMB control number

PETITION FOR EXTENSION OF TIME UNDER 37 CFR 1.136(a) **FY 2005** *(Fees pursuant to me Consolidated Appropriations Act, 2005 (H.R. 4818).)*	Docket Number 339000-000003
Application Number 11/477,807	Filed June 30, 2006
For METHODS AND APPARATUS FOR DEALING WITH MALWARE	
Art Unit 2432	Examiner C. Zecher

This is a request under the provisions of 37 CFR 1.136(a) to extend the period for filing a reply in the above identified application.

The requested extension and fee are as follows (check time period desired and enter the appropriate fee below):

		Fee	Fee Code	
☒	One Month (37 CFR 1.1 7(a)(1))	$130	1251	$130.00
☐	Two Month (37 CFR 1.1 7(a)(2))	$490	1252	
☐	Three Month (37 CFR 1.1 7(a)(3))	$1,110	1253	
☐	Four Month (37 CFR 1.1 7(a)(4))	$1730	1254	
☐	Five Month (37 CFR 1.1 7(a)(5))	$2,350	1255	

☐ Applicant claims small entity status. See 37 CFR 1.27.

☐ A check in the amount of the fee is enclosed.

☒ Payment by credit card. Form PTO-2038 is attached.

☐ The Director has already been authorized to charge fees in this application to a Deposit Account.

☐ The Director is hereby authorized to charge any fees which may be required, or credit any overpayment, to Deposit Account Number 50-3266. I have enclosed a duplicate copy of this sheet.

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

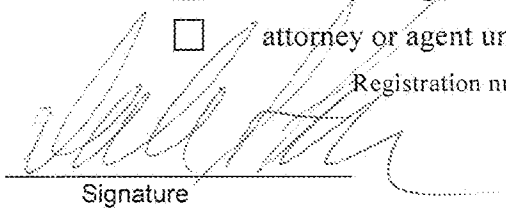
I am the

☐ applicant/inventor.

☐ assignee of record of the entire interest. See 37 CFR 3.71.
Statement under 37 CFR 3.73(b) is enclosed (Form PTO/SB/96).

☒ attorney or agent of record. Registration Number 28,872

☐ attorney or agent under 37 CFR 1.34.
Registration number if acting under 37 CFR 1.34 _____.



Signature

September 7, 2010
Date

Dale S. Lazar
Typed or printed name

703-773-4149
Telephone Number

NOTE: Signatures of all the inventors or assignees of record of the more interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below.

☒ Total of 1 forms are submitted

This collection of information is required by 37 CFR 1.136(a). The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S. C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 6 minutes to complete, including gathering, mating, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, US Patent and Trademark Office, U.S. Department OF Commerce, P.O. Box 1450, Alexandria, V.4 22313-4150. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patent. P.O. Box 1450, Alexandria, DA 22233-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2

American LegalNet, Inc
www.USCourtForms.com

Trend Micro, Inc.
EX1007-476

PTO/SB/06 (07-06)
Approved for use through 1/31/2007. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875	Application or Docket Number 11/477,807	Filing Date 06/30/2006	☐ To be Mailed

APPLICATION AS FILED – PART I

	(Column 1)	(Column 2)	SMALL ENTITY ☐		OR	OTHER THAN SMALL ENTITY	
FOR	NUMBER FILED	NUMBER EXTRA	RATE ($)	FEE ($)		RATE ($)	FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A			N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =		OR	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =			X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).						
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL			TOTAL	

APPLICATION AS AMENDED – PART II

AMENDMENT	09/07/2010	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE ($)	ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	* 45	Minus	** 47	= 0	X $ =		OR	X $52=	0
	Independent (37 CFR 1.16(h))	* 8	Minus	***8	= 0	X $ =		OR	X $220=	0
	☐ Application Size Fee (37 CFR 1.16(s))									
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	0

AMENDMENT		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)		RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =		OR	X $ =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =		OR	X $ =	
	☐ Application Size Fee (37 CFR 1.16(s))									
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.

** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".

*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".

The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Legal Instrument Examiner:
/BURNELL L. ROSS/

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application. Currently amended claims are shown with additions underlined and deletions in ~~strikethrough text~~ except double brackets may be placed before and after the deleted characters to show deletion of five or fewer characters.

1. (CURRENTLY AMENDED) A method of classifying a computer object as malware, the method comprising:

at a base computer, receiving data about a computer object from each of plural remote computers on which the object or similar objects are stored;

comparing in the base computer the data about the computer object received from the plural computers; [[and,]]

classifying the computer object as malware on the basis of said comparison;

when the computer object is not known malware, initially classifying an object as not malware;

generating a mask for said object that defines acceptable behaviour for the object; and

monitoring operation of the object on at least one of the remote computers and reclassifying the object as malware if the actual monitored behaviour extends beyond that permitted by the mask.

Trend Micro, Inc.
EX1007-478

2. (ORIGINAL) A method according to claim 1, wherein the data about the computer object that is sent from the plural remote computers to the base computer and that is used in the comparison includes one or more of: executable instructions contained within or constituted by the object; the size of the object; the current name of the object; the physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers.

3. (ORIGINAL) A method according to claim 1, wherein the comparing identifies relationships between the object and other objects.

4. (ORIGINAL) A method according to claim 3, wherein if at least one other object to which said object is related is classed as malware, then classifying said object as malware.

5. (ORIGINAL) A method according to claim 3, wherein said other objects include the object or similar objects stored on at least some of the remote computers.

6. (ORIGINAL) A method according to claim 3, wherein said other objects include other objects that are parent objects or child objects or otherwise process-related objects to said object.

7. (ORIGINAL) A method according to claim 1, wherein the data is sent in the form of key that is obtained by a hashing process carried out in respect of the objects on the respective remote computers.

8. (ORIGINAL) A method according to claim 7, wherein the key has at least one component that represents executable instructions contained within or constituted by the object.

9. (ORIGINAL) A method according to claim 7, wherein the key has at least one component that represents data about said object.

10. (ORIGINAL) A method according to claim 9, wherein said data about said object includes at least one of: the current name of the object; the physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers.

11. (ORIGINAL) A method according to claim 7, wherein the key has at least one component that represents the physical size of the object.

12. (CANCELLED)

13. (CURRENTLY AMENDED) Apparatus for classifying a computer object as malware, the apparatus comprising:

a base computer constructed and arranged to receive data about a computer object from each of plural remote computers on which the object or similar objects are stored;

the base computer being constructed and arranged to compare the data about the computer object received from said plural computers; and,

the base computer being constructed and arranged to classify the computer object as malware on the basis of said comparison,

wherein the base computer is constructed and arranged to generate a mask for an object that is initially classed not as malware, said mask defining acceptable behaviour for the object, and the base computer or each of the plural remote computers is constructed and arranged to monitor operation of the object on at least one of the remote computers and to reclassify the object as malware if the actual monitored behaviour extends beyond that permitted by the mask.

14. (ORIGINAL) Apparatus according to claim 13, wherein the base computer is constructed and arranged so as to compare the data where the data includes one or more of: executable instructions contained within or constituted by the object; the size of the object; the current name of the object; the physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers.

15. (ORIGINAL) Apparatus according to claim 13, wherein the base computer is constructed and arranged so that the comparing identifies relationships between the object and other objects.

16. (ORIGINAL) Apparatus according to claim 15, the base computer is constructed and arranged so that if at least one other object to which said object is related is classed as malware, then said object is classified as malware.

17. (ORIGINAL) Apparatus according to claim 15, wherein the base computer is constructed and arranged so that said other objects include the object or similar objects stored on at least some of said remote computers.

18. (ORIGINAL) Apparatus according to claim 15, wherein the base computer is constructed and arranged so that said other objects include other objects that are parent objects or child objects or otherwise process-related objects to said object.

19. (ORIGINAL) Apparatus according to claim 13, wherein the base computer is constructed and arranged so as to be able to process data that is sent in the form of key that is obtained by a hashing process carried out in respect of the objects on said respective remote computers.

20. (ORIGINAL) Apparatus according to claim 19, wherein the key has at least one component that represents executable instructions contained within or constituted by the object.

21. (ORIGINAL) Apparatus according to claim 19, wherein the key has at least one component that represents data about said object.

22. (ORIGINAL) Apparatus according to claim 21, wherein said data about said object includes at least one of: the current name of the object; the physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers.

23. (ORIGINAL) Apparatus according to claim 19, wherein the key has at least one component that represents the physical size of the object.

24. (CANCELLED)

25. (CURRENTLY AMENDED) A method of providing data about a computer object from a remote computer to a base computer so that a comparison can be made at the base computer with similar data received from other remote computers, the method comprising:

providing from a remote computer to a base computer data about a computer object that is stored on the remote computer;

the data including one or more of: executable instructions contained within or constituted by the object; the size of the object; the current name of the object; the physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers;

the data being sent in the form of key that is obtained by a hashing process carried out in respect of the object on the remote computer,

receiving at the remote computer a mask for a computer object stored at the remote computer, said mask defining acceptable behaviour for the object;

executing the computer object at the remote computer; and

monitoring operation of the computer object at the remote computer compared to the behaviour permitted by the mask and, in the event that the behaviour of the object extends beyond that permitted by the mask, sending data to the base computer that the behaviour of the object has exceeded that permitted by the mask.

26. (ORIGINAL) A method according to claim 25, wherein the key has at least one component that represents executable instructions contained within or constituted by the object.

27. (ORIGINAL) A method according to claim 25, wherein the key has at least one component that represents data about said object.

28. (ORIGINAL) A method according to claim 27, wherein said data about said object includes at least one of: the current name of the object; the physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers.

29. (ORIGINAL) A method according to claim 25, wherein the key has at least one component that represents the physical size of the object.

30. (CURRENTLY AMENDED) A carrier storing a computer program comprising program instructions for causing a computer to perform the method of claim 25.

31. (WITHDRAWN) A method of determining the protection that a remote computer has from malware, the method comprising:

receiving at a base computer details of all or selected security products operating at a point in time on said remote computer;

receiving similar information from other remote computers connected to the base computer; and,

identifying any malware processes that were not identified by said other remote computers having that particular combination of security products.

Trend Micro, Inc.
EX1007-485

32. (WITHDRAWN) A method according to claim 31, comprising providing information to the user of said remote computer that said remote computer may be susceptible to attack by said malware processes on the basis of said identifying.

33. (WITHDRAWN) A method according to claim 31, wherein the details of all or selected security products includes the name of the security products, versions, and loaded signature files.

34. (WITHDRAWN) Apparatus for determining the protection that a remote computer has from malware, the apparatus comprising:

a base computer constructed and arranged to receive computer details of all or selected security products operating at a point in time on said remote computer;

the base computer being constructed and arranged to receive similar information from other remote computers connected to the base computer; and,

the base computer being constructed and arranged to identify any malware processes that were not identified by said other remote computers having that particular combination of security products.

35. (WITHDRAWN) Apparatus according to claim 34, wherein the base computer is constructed and arranged to provide information to the user of said remote computer that said remote computer may be susceptible to attack by said malware processes on the basis of said identifying.

36. (WITHDRAWN) Apparatus according to claim 34, wherein the details of all or selected security products includes the name of the security products, versions, and loaded signature files.

37. (WITHDRAWN) A method of classifying a computer object as malware or as safe, wherein said computer object is a descendant or otherwise related object of a first computer object, the method comprising:

classifying a first computer object as malware or as safe;

identifying in a key relating to said first computer object a component that uniquely identifies the first computer object and that is inherited or otherwise present in the key of a descendant or other related computer object of the first computer object; and,

classifying said computer object as malware or as safe as the case may be on the basis of the unique identifier component being present in the key of said computer object.

38. (WITHDRAWN) A method according to claim 37, comprising monitoring activities of said first computer object and reclassifying the first computer object as malware in the case that it was initially classified as safe and subsequently determined to be malware, the method further comprising automatically classifying as malware any computer object that has a key in which said unique identifier component is present.

39. (WITHDRAWN) A computer program comprising program instructions for causing a computer to perform the method of claim 38.

40. (WITHDRAWN) Apparatus for classifying a computer object as malware or as safe, wherein said computer object is a descendant or otherwise related object of a first computer object, the apparatus comprising:

a computer constructed and arranged to classify a first computer object as malware or as safe;

the computer being constructed and arranged to identify in a key relating to said first computer object a component that uniquely identifies the first computer object and that is inherited or otherwise present in the key of a descendant or other related computer object of the first computer object; and,

the computer being constructed and arranged to classify said computer object as malware or as safe as the case may be on the basis of the unique identifier component being present in the key of said computer object.

41. (WITHDRAWN) Apparatus according to claim 40, wherein the computer is constructed and arranged to monitor activities of said first computer object and reclassify the first computer object as malware in the case that it was initially classified as safe and subsequently determined to be malware, the computer further being constructed and arranged to automatically classify as malware any computer object that has a key in which said unique identifier component is present.

42. (WITHDRAWN) A method of installing software on a computer, the method comprising:

on initiation of installation of software on a computer, providing a computer-generated prompt on the computer to a user to ascertain whether the user authorises the installation; and,

ceasing the installation if a user authorisation is not received, else:

receiving at the computer the user's authorisation to proceed with the installation;

proceeding with the installation;

obtaining data about computer objects that are created or used during the installation;

storing said data at the local computer.

43. (WITHDRAWN) A method according to claim 42, wherein the locally stored data is referred to during the installation to ensure that all objects created or used during the installation are part of the installation process, and, if it is found that objects created or used during the installation are not part of the installation process, either or both of: (i) ceasing the installation and (ii) providing a computer-generated prompt on the computer to the user accordingly.

44. (WITHDRAWN) A method according to claim 42, comprising connecting the computer to a community database that is connectable to a plurality of computers, and uploading the stored data to the community database for comparison with similar data provided by other computers.

45. (WITHDRAWN) A method according to claim 42, comprising downloading data about trusted installers to the computer, said data about trusted installers being referred to during the installation such that any objects relating to or created by the trusted installer are automatically authorised to proceed.

46. (WITHDRAWN) A method according to claim 45, wherein said data about trusted installers is referred to only for a predetermined time period following receipt at the computer of the user's authorisation to proceed with the installation.

47. (WITHDRAWN) A computer program comprising program instructions for causing a computer to perform the method of claim 42.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/477,807	06/30/2006	Melvyn Morris	339000-000003	4785

47604 7590 05/04/2010

DLA PIPER LLP US
P. O. BOX 2758
RESTON, VA 20195

EXAMINER	
ZECHER, CORDELIA P K	
ART UNIT	PAPER NUMBER
2432	

MAIL DATE	DELIVERY MODE
05/04/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Trend Micro, Inc.
EX1007-491

Office Action Summary	Application No.	Applicant(s)	
	11/477,807	MORRIS ET AL.	
	Examiner	Art Unit	
	Cordelia Zecher	2432	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *09 April 2010*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-30* is/are pending in the application.

4a) Of the above claim(s) *31-47* is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-30* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *30 June 2006* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

Trend Micro, Inc.
EX1007-492

DETAILED ACTION

Election/Restrictions

1. This application contains claims directed to the following patentably distinct species:

a. Species 1: A first aspect of the invention comparing data about the object received from plural computers and classifying the objects as malware on the basis of the comparisons (page 2, paragraphs 14-16). (claims 1-30).

b. Species 2: A fourth aspect of the invention, for determining the protection that a remote computer has from malware, by identifying any malware processes that were not identified (page 3, paragraphs 37-40). (claims 31-39).

c. Species 3: A sixth aspect of the invention, for identifying, in a key of a first computer object, a component that uniquely identifies the first computer object and that is inherited or otherwise present in the key of a descendant or other related computer object (page 3, paragraphs 48-51). (claims 40 and 41)

2. The species are independent or distinct because claims to the different species recite the mutually exclusive characteristics of such species. In addition, these species are not obvious variants of each other based on the current record.

3. Applicant is required under 35 U.S.C. 121 to elect a single disclosed species for prosecution on the merits to which the claims shall be restricted if no generic claim is finally held to be allowable. Currently, no claims are generic.

4. There is an examination and search burden for these patentably distinct species due to their mutually exclusive characteristics. The species require a different field of

search (e.g., searching different classes/subclasses or electronic resources, or employing different search queries); and/or the prior art applicable to one species would not likely be applicable to another species; and/or the species are likely to raise different non-prior art issues under 35 U.S.C. 101 and/or 35 U.S.C. 112, first paragraph.

5. **Applicant is advised that the reply to this requirement to be complete must include (i) an election of a species to be examined** even though the requirement may be traversed (37 CFR 1.143) **and (ii) identification of the claims encompassing the elected species**, including any claims subsequently added. An argument that a claim is allowable or that all claims are generic is considered nonresponsive unless accompanied by an election.

6. The election of the species may be made with or without traverse. To preserve a right to petition, the election must be made with traverse. If the reply does not distinctly and specifically point out supposed errors in the election of species requirement, the election shall be treated as an election without traverse. Traversal must be presented at the time of election in order to be considered timely. Failure to timely traverse the requirement will result in the loss of right to petition under 37 CFR 1.144. If claims are added after the election, applicant must indicate which of these claims are readable on the elected species.

7. Should applicant traverse on the ground that the species are not patentably distinct, applicant should submit evidence or identify such evidence now of record showing the species to be obvious variants or clearly admit on the record that this is the case. In either instance, if the examiner finds one of the species unpatentable over the

prior art, the evidence or admission may be used in a rejection under 35 U.S.C. 103(a) of the other species.

8. Upon the allowance of a generic claim, applicant will be entitled to consideration of claims to additional species which depend from or otherwise require all the limitations of an allowable generic claim as provided by 37 CFR 1.141.

9. Applicant is reminded that upon the cancellation of claims to a non-elected invention, the inventorship must be amended in compliance with 37 CFR 1.48(b) if one or more of the currently named inventors is no longer an inventor of at least one claim remaining in the application. Any amendment of inventorship must be accompanied by a request under 37 CFR 1.48(b) and by the fee required under 37 CFR 1.17(i).

10. Applicant's election without traverse of species 1 in the reply filed on April 9, 2010 is acknowledged.

Specification

The following guidelines illustrate the preferred layout for the specification of a utility application. These guidelines are suggested for the applicant's use.

Arrangement of the Specification

As provided in 37 CFR 1.77(b), the specification of a utility application should include the following sections in order. Each of the lettered items should appear in upper case, without underlining or bold type, as a section heading. If no text follows the section heading, the phrase "Not Applicable" should follow the section heading:

(a) TITLE OF THE INVENTION.
(b) CROSS-REFERENCE TO RELATED APPLICATIONS.
(c) STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT.
(d) THE NAMES OF THE PARTIES TO A JOINT RESEARCH AGREEMENT.
(e) INCORPORATION-BY-REFERENCE OF MATERIAL SUBMITTED ON A

COMPACT DISC.

(f) BACKGROUND OF THE INVENTION.

(1) Field of the Invention.

(2) Description of Related Art including information disclosed under 37 CFR 1.97 and 1.98.

(g) BRIEF SUMMARY OF THE INVENTION.

(h) BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWING(S).

(i) DETAILED DESCRIPTION OF THE INVENTION.

(j) CLAIM OR CLAIMS (commencing on a separate sheet).

(k) ABSTRACT OF THE DISCLOSURE (commencing on a separate sheet).

(l) SEQUENCE LISTING (See MPEP § 2424 and 37 CFR 1.821-1.825. A "Sequence Listing" is required on paper if the application discloses a nucleotide or amino acid sequence as defined in 37 CFR 1.821(a) and if the required "Sequence Listing" is not submitted as an electronic document on compact disc).

11. The disclosure is objected to because of the following informalities: The specification is not divided into the appropriate sections as listed above.

12. The lengthy specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

13. Appropriate correction is required.

Claim Rejections - 35 USC § 101

14. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

15. Claim 30 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The claim is directed to a computer program. A

computer program does not fall into one of the four statutory categories of invention and would be considered software per se. Therefore the claim is non-statutory.

Claim Rejections - 35 USC § 103

16. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

17. The factual inquiries set forth in *Graham* v. *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

18. Claims 1 – 9, 12 – 21, 24 – 27, and 30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dozortsev's US Publication 2003/0177394 A1 in view of Sheymov et al's US Publication 2001/0052014 A1.

19. Referring to claims 1 and 13, Dozortsev teaches:

a. At a base computer, receiving data about a computer object from a remote computer on which the object or similar objects are stored (page 5, paragraph 44).

b. Comparing in the base computer the data about the computer object received from the plural computers (page 5, paragraph 44).

c. Classifying the computer object as malware on the basis of said comparison (page 5, paragraph 44).

2. Dozortsev does not explicitly teach receiving data from plural remote computers. However, Sheymov teaches having both centralized and distributed databases of information so there is a greater chance of being able to react to the attempt in real time (page 2, paragraph 23). Dozortsev and Sheymov are from the same field of endeavor, network protection. At the time of invention, it would have been obvious to modify the database of Dozortsev to include the multiple databases of Sheymov. The suggestion for doing so would have been to have a greater chance to react to the attack in real time (page 2, paragraph 23).

3. Referring to claims 2 and 14, Dozortsev teaches that the data about the computer object that is sent from the remote computer to the base computer and that is used in the comparison includes executable instructions contained within or constituted by the subject (page 3, paragraph 21).

4. Referring to claims 3 and 15, Dozortsev teaches comparing identifies relationships between the object and other objects (page 3, paragraph 22).

5. Referring to claims 4 and 16, Dozortsev teaches wherein if at least one other object to which said object is related is classified as malware, then classifying said object as malware (page 3, paragraph 25).

6. Referring to claims 5 and 17, Dozortsev teaches the other objects include objects stored on at least some of the remote computers (page 5, paragraphs 43-44).

7. Referring to claims 6 and 18, Dozortsev teaches that the other objects include otherwise process related objects (page 5, paragraph 44).

8. Referring to claims 7 and 19, Dozortsev teaches that the data is sent in the form of a key that is obtained by a hashing process carried out in respect of the objects on the remote computer (page 3, paragraph 21).

9. Referring to claims 8 and 20, Dozortsev teaches that the key has one component that represents executable instructions contained within or constituted by the object (page 3, paragraph 21).

10. Referring to claims 9 and 21, Dozortsev teaches that the key has at least one component that represents data about said object (page 3, paragraph 21).

11. Referring to claims 12 and 24, Dozortsev teaches initially classifying an object as not malware, generating a mask for said object that defines acceptable behavior for the object, and comprising monitoring operation of the object on at least one of the remote computers and reclassifying the object as malware if the actual monitored behavior extends beyond that permitted by the mask (page 4, paragraphs 37-40).

12. Referring to claims 25 and 30, Dozortsev teaches:

d. Providing from a remote computer to a base computer data about a computer object that is stored on the remote computer (page 5, paragraph 44).

e. The data including executable instructions contained within or constituted by the object (page 3, paragraph 21).

f. The data being sent in the form of key that is obtained by a hashing process carried out in respect of the object on the remote computer (page 3, paragraph 21).

13. Dozortsev does not explicitly teach receiving data from plural remote computers. However, Sheymov teaches having both centralized and distributed databases of information so there is a greater chance of being able to react to the attempt in real time (page 2, paragraph 23). Dozortsev and Sheymov are from the same field of endeavor, network protection. At the time of invention, it would have been obvious to modify the database of Dozortsev to include the multiple databases of Sheymov. The suggestion for doing so would have been to have a greater chance to react to the attack in real time (page 2, paragraph 23).

14. Referring to claim 26, Dozortsev teaches that the key has one component that represents executable instructions contained within or constituted by the object (page 3, paragraph 21).

15. Referring to claim 27, Dozortsev teaches that the key has at least one component that represents data about said object (page 3, paragraph 21).

16. Claims 10, 11, 22, 23, 28, and 29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dozortsev in view of Sheymov as applied above, and further in view of Barton et al's US Publication 2005/0021994 A1. Dozortsev in view of Sheymov discloses all the limitations of the parent claims. Dozortsev in view of Sheymov does not explicitly disclose the data about the object including the name of the object, the

location or the size of the object. However, Barton discloses indicating for the object the file name, size and location (pages 2-3, paragraph 39). Dozortsev in view of Sheymov and B are analogous art because they are from the same field of endeavor, malware detection. At the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teachings of Dozortsev in view of Sheymov and Barton before him or her, to modify the signature of Dozortsev in view of Sheymov to include the data name and size of Barton. The suggestion/motivation for doing so would have been the characteristics are strongly indicative of a particular file being one of the candidates (page 1, paragraph 14).

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Cordelia Zecher whose telephone number is (571)272-7771. The examiner can normally be reached on Monday - Thursday 8:00 - 5:00 EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Gilberto Barron can be reached on 571-272-3799. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/C. Z./
Examiner, Art Unit 2432

/Benjamin E Lanier/
Primary Examiner, Art Unit 2432

Notice of References Cited	Application/Control No. 11/477,807	Applicant(s)/Patent Under Reexamination MORRIS ET AL.	
	Examiner Cordelia Zecher	Art Unit 2432	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-2001/0052014 A1	12-2001	Sheymov et al.	709/225
*	B	US-2003/0177394 A1	09-2003	Dozortsev, Dmitri	713/201
*	C	US-2005/0021994 A1	01-2005	Barton et al.	713/200
	D	US-			
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11477807	MORRIS ET AL.
	Examiner	Art Unit
	Cordelia Zecher	2432

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	04/28/2010								
	1	✓								
	2	✓								
	3	✓								
	4	✓								
	5	✓								
	6	✓								
	7	✓								
	8	✓								
	9	✓								
	10	✓								
	11	✓								
	12	✓								
	13	✓								
	14	✓								
	15	✓								
	16	✓								
	17	✓								
	18	✓								
	19	✓								
	20	✓								
	21	✓								
	22	✓								
	23	✓								
	24	✓								
	25	✓								
	26	✓								
	27	✓								
	28	✓								
	29	✓								
	30	✓								
	31	N								
	32	N								
	33	N								
	34	N								
	35	N								
	36	N								

Trend Micro, Inc.
EX1007-504

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11477807	MORRIS ET AL.
	Examiner	**Art Unit**
	Cordelia Zecher	2432

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	04/28/2010								
	37	N								
	38	N								
	39	N								
	40	N								
	41	N								
	42	N								
	43	N								
	44	N								
	45	N								
	46	N								
	47	N								

Trend Micro, Inc.
EX1007-505

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11477807	MORRIS ET AL.
	Examiner	Art Unit
	Cordelia Zecher	2432

SEARCHED			
Class	Subclass	Date	Examiner
726	24	4/28/10	cz

SEARCH NOTES		
Search Notes	Date	Examiner
Searched EAST (see attached)	4/28/10	cz
Inventor Search	4/28/10	cz

INTERFERENCE SEARCH			
Class	Subclass	Date	Examiner

Trend Micro, Inc.
EX1007-506

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

BIB DATA SHEET

CONFIRMATION NO. 4785

SERIAL NUMBER	FILING or 371(c) DATE RULE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
11/477,807	06/30/2006	726	2432	339000-000003

APPLICANTS

Melvyn Morris, Turnditch, UNITED KINGDOM;
Paul Stubbs, Wyboston, UNITED KINGDOM;
Markus Hartwig, Milton Keynes, UNITED KINGDOM;
Darren Harter, Hucclecote, UNITED KINGDOM;

**** CONTINUING DATA ***************************

**** FOREIGN APPLICATIONS ***************************

UNITED KINGDOM 0513375.6 06/30/2005

**** IF REQUIRED, FOREIGN FILING LICENSE GRANTED ****

07/26/2006

Foreign Priority claimed	35 USC 119(a-d) conditions met	Verified and Acknowledged	Met after Allowance	STATE OR COUNTRY	SHEETS DRAWINGS	TOTAL CLAIMS	INDEPENDENT CLAIMS
☑ Yes ☐ No	☑ Yes ☐ No	/CORDELIA P K ZECHER/ Examiner's Signature	☐ Initials	UNITED KINGDOM	3	47	8

ADDRESS

DLA PIPER LLP US
P. O. BOX 2758
RESTON, VA 20195
UNITED STATES

TITLE

Methods and apparatus for dealing with malware

FILING FEE RECEIVED		
3480	FEES: Authority has been given in Paper No._____ to charge/credit DEPOSIT ACCOUNT No._____ for following:	☐ All Fees ☐ 1.16 Fees (Filing) ☐ 1.17 Fees (Processing Ext. of time) ☐ 1.18 Fees (Issue) ☐ Other _____ ☐ Credit

Trend Micro, Inc.
EX1007-507

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	1	("20060026684").PN.	US-PGPUB; USPAT	OR	OFF	2010/04/28 08:32
L2	1	1 and malware	US-PGPUB; USPAT	OR	ON	2010/04/28 08:33
L3	1164	726/24.ccls.	US-PGPUB; USPAT	OR	ON	2010/04/28 08:33
L6	2821	726/23-25.ccls.	US-PGPUB; USPAT	OR	ON	2010/04/28 08:34
L7	6	(("6772346") or ("6944772") or ("20040006704") or ("20040073810") or ("20050021994") or ("20040153644")).PN.	US-PGPUB; USPAT	OR	OFF	2010/04/28 09:18
L8	1	("20030177394").PN.	US-PGPUB; USPAT	OR	OFF	2010/04/28 10:48
L9	0	central near database with each with network	US-PGPUB; USPAT	OR	ON	2010/04/28 12:51
L10	0	database with each with network	US-PGPUB; USPAT	OR	ON	2010/04/28 12:51
L11	115859	database with network	US-PGPUB; USPAT	OR	ON	2010/04/28 12:52
L12	2632	central near database with network	US-PGPUB; USPAT	OR	ON	2010/04/28 12:52
L13	30	central near database with network with separate	US-PGPUB; USPAT	OR	ON	2010/04/28 12:53
L14	17	central near database with distribute with network	US-PGPUB; USPAT	OR	ON	2010/04/28 12:53

L15	199	central near database with distribut$3 with network	US-PGPUB; USPAT	OR	ON	2010/04/28 12:54
L16	0	central near database with distribute with network with known	US-PGPUB; USPAT	OR	ON	2010/04/28 12:54
L17	4	central near database with distribut$3 with network with known	US-PGPUB; USPAT	OR	ON	2010/04/28 12:55
L18	240	database with distribut$3 with network with known	US-PGPUB; USPAT	OR	ON	2010/04/28 12:55
L19	24	database with distribut$3 with network with known and (virus or malware)	US-PGPUB; USPAT	OR	ON	2010/04/28 12:56
S1	1	"20070016953"	US-PGPUB; USPAT	OR	ON	2010/01/07 08:58

4/ 28/ 2010 1:24:47 PM
C:\ Documents and Settings\ ckane\ My Documents\ EAST\ Workspaces\ 11477807.wsp

Receipt date: 01/12/2007 11477807 - GAU: 2432

OIPE TAP 56
JAN 12 2007
PATENT & TRADEMARK OFFICE

Docket Number: 011765-0355986
Client Reference: AF/JG/P10717US

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re the Application of

MELVYN MORRIS, et al. Group Art Unit: 2131

Application No.: 11/477,807 Examiner: Unassigned

Filed: June 30, 2006 Confirmation No.: 4785

For: METHODS AND APPARATUS FOR DEALING WITH MALWARE

INFORMATION DISCLOSURE STATEMENT

Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

Sir:

Pursuant to 37 CFR 1.56, the attention of the Patent and Trademark Office is hereby directed to the references listed on the attached PTO-1449. Copies of the required references are provided herewith. It is respectfully requested that the information be expressly considered during the prosecution of this application, and that the references be made of record therein and appear among the "References Cited" on any patent to issue therefrom. Applicants respectfully request the Examiner return an initialed copy of the enclosed Form PTO-1449 to Applicants with the next Office communication to indicate that the references have been considered, per MPEP § 609.

This Information Disclosure Statement is being within three months of the mailing date of the International Search Report in the related international application. No fee is required.

Respectfully Submitted,
PILLSBURY WINTHROP SHAW PITTMAN LLP

Christine H. McCarthy
Registration Number 41844
Customer Number: 00909

Date: January 12, 2007
Telephone: (703) 770-7900
Facsimile: (703) 770-7901
P.O. Box 10500
McLean, VA 22102

/Cordelia Zecher/ 04/28/2010

4005/2805v1

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.Z./

Trend Micro, Inc.
EX1007-510

Receipt date: 01/18/2007

PTO JAN 18 2007 PATENT & TRADEMARK OFFICE

11477807 - GAU: 2432

FORM PTO-1449 (modified)
To: U.S. Department of Commerce
(PW FORM PAT-1449) Patent and Trademark Office

Atty. Dkt. No.	M#	Client Ref.
	11765-0355986	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

Applicant: MORRIS et al.
Appln. No.: 11/477,807
Filing Date: June 30, 2006
Examiner: Unassigned | Group Art Unit: 2131

Date: January 18, 2007 Page 1 of 1

U.S. PATENT DOCUMENTS

Examiner's Initials*		Document Number	Date MM/YYYY	Name (Family Name of First Inventor)	Class	Sub Class	Filing Date (if appropriate)
	AR	6,772,346	08/2004	CHESS			
	BR	6,944,772	09/2005	DOZORTSEV			
	CR	2004/0006704	01/2004	DAHLSTROM			
	DR	2004/0073810	05/2004	RACINE			
	ER	2005/0021994	01/2005	BARTON			
	FR	2004/0153644	08/2004	McCORKENDALE			
	GR						
	HR						
	IR						
	JR						
	KR						
	LR						
	MR						
	NR						

FOREIGN PATENT DOCUMENTS

		Document Number	Date MM/YYYY	Country	Inventor Name		English Abstract		Translation Readily Available	
							Enclosed	No	Enclose	No
	OR	1 280 040	01/2003	EP						
	PR	1 549 012	12/2003	EP						
	QR	03/021402	03/2003	WO						
	RR	2004/097602	11/2004	WO						
	SR	02/33525	04/2002	WO						

OTHER (Including in this order Author, Title, Periodical Name, Date, Pertinent Pages, etc.)

	TR				
	UR				
	VR				
	WR				

Examiner /Cordelia Zecher/ Date Considered: 04/28/2010

*EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP § 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to Applicant.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH./C.Z./

OIPE IAP44
OCT 31 2006
PATENT & TRADEMARK OFFICE

PTO/SB/08a (07-05)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Substitute for form 1449A/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 1 of 1

Complete if Known	
Application Number	11/477,807
Filing Date	06/30/2006
First Named Inventor	MELVYN MORRIS
Art Unit	2131
Examiner Name	
Attorney Docket Number	011765-0355986

U. S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document Country Code[3] Number[4] Kind Code[5] (if known)			Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T[6]
		WO	02/33525	A2	04/25/2002			
		WO	03/021402	A2	03/13/2003			
		WO	2004/097602	A2	11/11/2004			
		EP	1 280 040	A3	03/24/2004			
		EP	1 549 012	A1	06/29/2005			

Examiner Signature	/Cordelia Zecher/	Date Considered	04/28/2010

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: **Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH./C.Z./

DOCKET NO. 339000-000003 PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of:	MORRIS, Melvyn e tal.	Confirmation Number: 4785
Application No.	11/477,807	Group Art Unit: 2432
Filed:	June 30, 2006	Examiner: C. P. Kane
For:	METHODS AND APPARATUS FOR DEALING WITH MALWARE	

RESPONSE TO RESTRICTION REQUIREMENT

Assistant Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

April 9, 2010

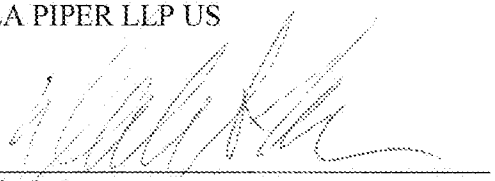
Dear Sir:

In response to the Office Action dated January 13, 2010, Applicants elect Group I, claims 1 to 41, without traverse. Applicants also wish to elect Species I directed to claims 1 to 30, 40 and 41. Applicants reserve the right to file a divisional application on the non-elected claims.

Please charge any shortage in the fees or credit any overpayment under 37 C.F.R. 1.17(a) – (d) to Deposit Account No. 50-3266.

Respectfully submitted,

DLA PIPER LLP US



Dale S. Lazar
Registration No. 28,872
Attorney for Applicant

DSL/jcc
PO Box 2758
Reston, VA 20195
(703) 773.4149 Telephone
(703) 773.5200 Fax

Trend Micro, Inc.
EX1007-513

Electronic Patent Application Fee Transmittal

Application Number:	11477807
Filing Date:	30-Jun-2006
Title of Invention:	Methods and apparatus for dealing with malware
First Named Inventor/Applicant Name:	Melvyn Morris
Filer:	Dale S. Lazar/Josefina Cheng
Attorney Docket Number:	339000-000003

Filed as Large Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Extension - 2 months with $0 paid	1252	1	490	490

Trend Micro, Inc.
EX1007-514

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
			Total in USD ($)	**490**

Electronic Acknowledgement Receipt

EFS ID:	7387595
Application Number:	11477807
International Application Number:	
Confirmation Number:	4785
Title of Invention:	Methods and apparatus for dealing with malware
First Named Inventor/Applicant Name:	Melvyn Morris
Customer Number:	47604
Filer:	Dale S. Lazar/Josefina Cheng
Filer Authorized By:	Dale S. Lazar
Attorney Docket Number:	339000-000003
Receipt Date:	09-APR-2010
Filing Date:	30-JUN-2006
Time Stamp:	16:03:53
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Credit Card
Payment was successfully received in RAM	$490
RAM confirmation Number	2401
Deposit Account	
Authorized User	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)

Trend Micro, Inc.
EX1007-516

1	Miscellaneous Incoming Letter	339000-3Transmittal.pdf	351546 61d66b624205ed8b2d8642b280e3ab13a1f34367	no	1
Warnings:					
Information:					
2	Extension of Time	339000-3Extension.pdf	481137 7e4f3f4eb8fd0c5f45854a43f103979bf5bfda83	no	1
Warnings:					
Information:					
3	Response to Election / Restriction Filed	339000-3RR.pdf	272870 ae31c051bb53962c8e31214854150661590b709e	no	1
Warnings:					
Information:					
4	Fee Worksheet (PTO-875)	fee-info.pdf	30037 205ed9cea35697acd607dfde935969f7ae5c5550	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	1135590		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

DOCKET NO.: 339000-000003 PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Honorable Commissioner for Patents
Post Office Box 1450
Alexandria, Virginia 22313-1450

Re: Application No.: 11/477,807
Applicants: MORRIS, Melvyn et al.
Filing Date June 30, 2006
For: METHODS AND APPARATUS FOR DEALING WITH MALWARE

Dear Sir:

Attached hereto for filing are the following papers. The fee has been calculated as shown below.

☒ Response to the Office Action dated January 13, 2010; and

☒ Request for Extension of Time Two Months.

	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR		PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)
Total	47	- 47		=	x $52	
Independent	8	- 8		=	x $220	
First Presentation of Multiple Dependent Claim						$0.00
Two Months Extension of Time						$490.00
					Total Fees	$490.00

Our payment in the amount of **$490.00** covering any required fees is attached hereto. In the event any variance exists between the amount enclosed and the Patent Office charges for filing the above-noted documents, including any fees required under 37 C.F.R. 1.136 for any necessary extension of time to make the filing of the attached documents timely, please charge or credit the difference to Deposit Account No. 50-3266. Further, if these papers are not considered timely filed, then a request is hereby made under 37 C.F.R. 1.136 for the necessary extension of time.

Respectfully submitted,

DLA PIPER LLP US

Dale S. Lazar
Registration No. 28,872

Dated: April 9, 2010

Trend Micro, Inc.
EX1007-518

PTO/SB/22 (10-04)
Approved for use through 7/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE
Under the paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

PETITION FOR EXTENSION OF TIME UNDER 37 CFR 1.136(a) FY 2005 *(Fees pursuant to the Consolidated Appropriations Act, 2005 (H.R. 4818).)*	Docket Number 339000-000003
Application Number 11/477,807	Filed June 30, 2006
For METHODS AND APPARATUS FOR DEALING WITH MALWARE	
Art Unit 2432	Examiner C. P. Kane

This is a request under the provisions of 37 CFR 1.136(a) to extend the period for filing a reply in the above identified application.

The requested extension and fee are as follows (check time period desired and enter the appropriate fee below):

		Fee	Fee Code	
☐	One Month (37 CFR 1.1 7(a)(1))	$130	1251	
☒	Two Month (37 CFR 1.1 7(a)(2))	$490	1252	$490.00
☐	Three Month (37 CFR 1.1 7(a)(3))	$1,110	1253	
☐	Four Month (37 CFR 1.1 7(a)(4))	$1730	1254	
☐	Five Month (37 CFR 1.1 7(a)(5))	$2,350	1255	

☐ Applicant claims small entity status. See 37 CFR 1.27.

☐ A check in the amount of the fee is enclosed.

☒ Payment by credit card. Form PTO-2038 is attached.

☐ The Director has already been authorized to charge fees in this application to a Deposit Account.

☐ The Director is hereby authorized to charge any fees which may be required, or credit any overpayment, to Deposit Account Number 50-3266. I have enclosed a duplicate copy of this sheet.

WARNING: Information **on this form may** become **public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.**

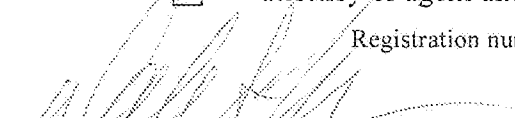
I am the

☐ applicant/inventor.

☐ assignee of record of the entire interest. See 37 CFR 3.71.
Statement under 37 CFR 3.73(b) is enclosed (Form PTO/SB/96).

☒ attorney or agent of record. Registration Number 28,872

☐ attorney or agent under 37 CFR 1.34.
Registration number if acting under 37 CFR 1.34 _____.



Signature

April 9, 2010
Date

Dale S. Lazar
Typed or printed name

703-773-4149
Telephone Number

NOTE: Signatures of all the inventors or assignees of record of the more interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below.

☒ Total of 1 forms are submitted

This collection of information is required by 37 CFR 1.136(a). The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S. C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 6 minutes to complete, including gathering, mating, and submitting the completed application form to the USPTO. Time will vary depending upon the Individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, US Patent and Trademark Office, U.S. Department OF Commerce, P.O. Box 1450, Alexandria, V.4 22313-4150. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patent, P.O. Box 1450, Alexandria, DA 22233-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2

American LegalNet, Inc
www.USCourtForms.com

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/477,807	06/30/2006	Melvyn Morris	339000-000003	4785

47604 7590 01/13/2010

DLA PIPER LLP US
P. O. BOX 2758
RESTON, VA 20195

EXAMINER	
KANE, CORDELIA P	
ART UNIT	**PAPER NUMBER**
2432	
MAIL DATE	**DELIVERY MODE**
01/13/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Trend Micro, Inc.
EX1007-520

Office Action Summary	Application No.	Applicant(s)	
	11/477,807	MORRIS ET AL.	
	Examiner	Art Unit	
	CORDELIA KANE	2432	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *1* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *30 June 2006*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-47* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☐ Claim(s) _____ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☒ Claim(s) *1-47* are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

Trend Micro, Inc.
EX1007-521

DETAILED ACTION

Election/Restrictions

1. Restriction to one of the following inventions is required under 35 U.S.C. 121:

I. Claims 1 – 41, drawn to classifying objects.

II. Claims 42 – 47, drawn to installing software on a computer.

The inventions are distinct, each from the other because of the following reasons:

2. Inventions I, and II are directed to related processes. The related inventions are distinct if: (1) the inventions as claimed are either not capable of use together or can have a materially different design, mode of operation, function, or effect; (2) the inventions do not overlap in scope, i.e., are mutually exclusive; and (3) the inventions as claimed are not obvious variants. See MPEP § 806.05(j). In the instant case, the inventions have different functions. Invention I classifies objects as malware. Invention II installs software. Furthermore, the inventions as claimed do not encompass overlapping subject matter and there is nothing of record to show them to be obvious variants.

3. Restriction for examination purposes as indicated is proper because all these inventions listed in this action are independent or distinct for the reasons given above and there would be a serious search and examination burden if restriction were not required because one or more of the following reasons apply:

(a) the inventions have acquired a separate status in the art in view of their different classification;

(b) the inventions have acquired a separate status in the art due to their recognized divergent subject matter;

(c) the inventions require a different field of search (for example, searching different classes/subclasses or electronic resources, or employing different search queries);

(d) the prior art applicable to one invention would not likely be applicable to another invention;

(e) the inventions are likely to raise different non-prior art issues under 35 U.S.C. 101 and/or 35 U.S.C. 112, first paragraph.

Applicant is advised that the reply to this requirement to be complete must include (i) an election of a invention to be examined even though the requirement may be traversed (37 CFR 1.143) **and (ii) identification of the claims encompassing the elected invention**.

The election of an invention may be made with or without traverse. To reserve a right to petition, the election must be made with traverse. If the reply does not distinctly and specifically point out supposed errors in the restriction requirement, the election shall be treated as an election without traverse. Traversal must be presented at the time of election in order to be considered timely. Failure to timely traverse the requirement will result in the loss of right to petition under 37 CFR 1.144. If claims are added after the election, applicant must indicate which of these claims are readable on the elected invention.

If claims are added after the election, applicant must indicate which of these claims are readable upon the elected invention.

Should applicant traverse on the ground that the inventions are not patentably distinct, applicant should submit evidence or identify such evidence now of record showing the inventions to be obvious variants or clearly admit on the record that this is the case. In either instance, if the examiner finds one of the inventions unpatentable over the prior art, the evidence or admission may be used in a rejection under 35 U.S.C. 103(a) of the other invention.

4. If Invention I is elected, this invention contains claims directed to the following patentably distinct species:

a. Species 1: A first aspect of the invention comparing data about the object received from plural computers and classifying the objects as malware on the basis of the comparisons (page 2, paragraphs 14-16). (claims 1-30, 40, 41).

b. Species 2: A fourth aspect of the invention, for determining the protection that a remote computer has from malware, by identifying any malware processes that were not identified (page 3, paragraphs 37-40). (claims 31-39).

The species are independent or distinct because claims to the different species recite the mutually exclusive characteristics of such species. In addition, these species are not obvious variants of each other based on the current record.

Applicant is required under 35 U.S.C. 121 to elect a single disclosed species for prosecution on the merits to which the claims shall be restricted if no generic claim is finally held to be allowable.

There is an examination and search burden for these patentably distinct species due to their mutually exclusive characteristics. The species require a different field of search (e.g., searching different classes/subclasses or electronic resources, or employing different search queries); and/or the prior art applicable to one species would not likely be applicable to another species; and/or the species are likely to raise different non-prior art issues under 35 U.S.C. 101 and/or 35 U.S.C. 112, first paragraph.

Applicant is advised that the reply to this requirement to be complete must include (i) an election of a species to be examined even though the requirement may be traversed (37 CFR 1.143) **and (ii) identification of the claims encompassing the elected species**, including any claims subsequently added. An argument that a claim is allowable or that all claims are generic is considered nonresponsive unless accompanied by an election.

The election of the species may be made with or without traverse. To preserve a right to petition, the election must be made with traverse. If the reply does not distinctly and specifically point out supposed errors in the election of species requirement, the election shall be treated as an election without traverse. Traversal must be presented at the time of election in order to be considered timely. Failure to timely traverse the requirement will result in the loss of right to petition under 37 CFR 1.144. If claims are added after the election, applicant must indicate which of these claims are readable on the elected species.

Should applicant traverse on the ground that the species are not patentably distinct, applicant should submit evidence or identify such evidence now of record

showing the species to be obvious variants or clearly admit on the record that this is the case. In either instance, if the examiner finds one of the species unpatentable over the prior art, the evidence or admission may be used in a rejection under 35 U.S.C. 103(a) of the other species.

Upon the allowance of a generic claim, applicant will be entitled to consideration of claims to additional species which depend from or otherwise require all the limitations of an allowable generic claim as provided by 37 CFR 1.141.

5. Applicant is reminded that upon the cancellation of claims to a non-elected invention, the inventorship must be amended in compliance with 37 CFR 1.48(b) if one or more of the currently named inventors is no longer an inventor of at least one claim remaining in the application. Any amendment of inventorship must be accompanied by a request under 37 CFR 1.48(b) and by the fee required under 37 CFR 1.17(i).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to CORDELIA KANE whose telephone number is (571)272-7771. The examiner can normally be reached on Monday - Thursday 8:00 - 5:00 EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Gilberto Barron can be reached on 571-272-3799. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/C. K./
Examiner, Art Unit 2432

/Benjamin E Lanier/
Primary Examiner, Art Unit 2432

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371 (c) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
11/477,807	06/30/2006	Melvyn Morris	011765-0355986

CONFIRMATION NO. 4785

OC000000023928527

909
PILLSBURY WINTHROP SHAW PITTMAN, LLP
P.O. BOX 10500
MCLEAN, VA 22102

Date Mailed: 05/17/2007

NOTICE REGARDING CHANGE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 05/02/2007.

- The Power of Attorney to you in this application has been revoked by the assignee who has intervened as provided by 37 CFR 3.71. Future correspondence will be mailed to the new address of record(37 CFR 1.33).

Office of Initial Patent Examination (571) 272-4000, or 1-800-PTO-9199
OFFICE COPY

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371 (c) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
11/477,807	06/30/2006	Melvyn Morris	339000-P0003

CONFIRMATION NO. 4785

OC000000023928564

47604
DLA PIPER US LLP
P. O. BOX 9271
RESTON, VA 20195

Date Mailed: 05/17/2007

NOTICE OF ACCEPTANCE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 05/02/2007.

The Power of Attorney in this application is accepted. Correspondence in this application will be mailed to the above address as provided by 37 CFR 1.33.

Office of Initial Patent Examination (571) 272-4000, or 1-800-PTO-9199
OFFICE COPY

PTO/SB/81 (01-06)
Approved for use through 12/31/2008. OMB 0651-0035
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

POWER OF ATTORNEY and CORRESPONDENCE ADDRESS INDICATION FORM

Application Number	11/477,807
Filing Date	June 30, 2006
First Named Inventor	Melvyn MORRIS et al.
Art Unit	Unknown
Examiner Name	Unknown
Attorney Docket Number	339000-P0003

I hereby revoke all previous powers of attorney given in the above-identified application.

I hereby appoint:

☒ Practitioners associated with the Customer Number: 47604

OR

☐ Practitioner(s) name below:

Name	Registration Number

as my/our attorney(s) or agent(s) to prosecute the application identified above, and to transact all business in the United States Patent and Trademark Office connected therewith.

Please recognize or change the correspondence address for the above-identified application to:

☒ The address associated with the above-mentioned Customer Number: 47604

OR

☐ The address associated with Customer Number:

OR

☐ Firm or individual name			
Address			
City		State	Zip
Country			
Telephone		Email	

I am the:

☐ Applicant/Inventor.

☒ Assignee of record of the entire interest. See 37 CFR 3.71.

Statement under 37 CFR 3.73(b) is enclosed. (Form PTO/SB/96)

SIGNATURE of Applicant or Assignee of Record

Signature		**Date**	2/26/07
Name	Paul Stubbs	**Telephone**	
Title and Company	Director, Trend Ltd.		

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below*

☒ ***Total of 1 forms are submitted.**

This collection of information is required by 37 CFR 1.31,1.32 and 1.33. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 9 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Trend Micro, Inc.
EX1007-530

PTO/SB/96 (12-06)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

STATEMENT UNDER 37 CFR 3.73(b)

Applicant/Patent Owner: Prevx Limited

Application No./Patent No./Control No.: 11/477,807 Filed/Issue Date: June 30, 2006

Entitled: METHODS AND APPARATUS FOR DEALING WITH MALWARE

Prevx Limited, a Corporation
(Name of Assignee) (Type of Assignee: corporation, partnership, university, government agency, etc.)

States that it is:

1. ☒ the assignee of the entire right, title, and interest; or

2. ☐ an assignee of less than the entire right, title and interest
(The extent (by percentage) of its ownership interest is _____%

in the patent application/ patent identified above by virtue of either:

A. ☒ An assignment from the inventor(s) of the patent application/patent identified above. The assignment was recorded in the United States Patent and Trademark Office at Reel 018348, Frame 0840, or a true copy of the original assignment is attached.

OR

B. ☐ A chain of title from the inventor(s), of the present patent application/ patent identified above, to the current assignee as follows:

1. From: _____ To: _____
The document was recorded in the United States Patent and trademark Office at Reel _____, Frame _____, or for which a copy thereof is attached.

2. From: _____ To: _____
The document was recorded in the United States Patent and trademark Office at Reel _____, Frame _____, or for which a copy thereof is attached.

3. From: _____ To: _____
The document was recorded in the United States Patent and trademark Office at Reel _____, Frame _____, or for which a copy thereof is attached.

☐ Additional documents in the chain of title are listed on a supplemental sheet.

As required by 37 CFR 3.73(b)(1)(i), the documentary evidence of the chain of title from the original owner to the assignee was, or concurrently is being, submitted for recordation pursuant to 37 CFR 3.11.

[NOTE: A separate copy (*i.e.*, a true copy of the original assignment document(s)) must be submitted to Assignment Division in accordance with 37 CFR Part 3, to record the assignment in the records of the USPTO. See MPEP 302.08]

The undersigned (whose title is supplied below) is authorized to act on behalf of the assignees.

[signature]	26/3/07
Signature	Date
Paul Sparks	
Printed or Typed Name	Telephone Number
Director	
Title	

This collection of information is required by 37 CFR 3.73(b). The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

REST1\593015.1

Trend Micro, Inc.
EX1007-531

Electronic Acknowledgement Receipt

EFS ID:	1735854
Application Number:	11477807
International Application Number:	
Confirmation Number:	4785
Title of Invention:	Methods and apparatus for dealing with malware
First Named Inventor/Applicant Name:	Melvyn Morris
Customer Number:	909
Filer:	Dale S. Lazar/Margaret Filipowicz
Filer Authorized By:	Dale S. Lazar
Attorney Docket Number:	011765-0355986
Receipt Date:	02-MAY-2007
Filing Date:	30-JUN-2006
Time Stamp:	09:52:17
Application Type:	Utility

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)	Multi Part /.zip	Pages (if appl.)
1	Power of Attorney	339000P0003POA.pdf	702308	no	1
Warnings:					

Information:					
2	Assignee showing of ownership per 37 CFR 3.73(b).	339000P0003StatementA.pdf	671580	no	1
Warnings:					
Information:					
		Total Files Size (in bytes):	1373888		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(c) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
11/477,807	06/30/2006	Melvyn Morris	011765-0355986

CONFIRMATION NO. 4785

909
PILLSBURY WINTHROP SHAW PITTMAN, LLP
P.O. BOX 10500
MCLEAN, VA22102

Title: Methods and apparatus for dealing with malware

Publication No. US-2007-0016953-A1
Publication Date: 01/18/2007

NOTICE OF PUBLICATION OF APPLICATION

The above-identified application will be electronically published as a patent application publication pursuant to 37 CFR 1.211, et seq. The patent application publication number and publication date are set forth above.

The publication may be accessed through the USPTO's publically available Searchable Databases via the Internet at www.uspto.gov. The direct link to access the publication is currently http://www.uspto.gov/patft/.

The publication process established by the Office does not provide for mailing a copy of the publication to applicant. A copy of the publication may be obtained from the Office upon payment of the appropriate fee set forth in 37 CFR 1.19(a)(1). Orders for copies of patent application publications are handled by the USPTO's Office of Public Records. The Office of Public Records can be reached by telephone at (703) 308-9726 or (800) 972-6382, by facsimile at (703) 305-8759, by mail addressed to the United States Patent and Trademark Office, Office of Public Records, Alexandria, VA 22313-1450 or via the Internet.

In addition, information on the status of the application, including the mailing date of Office actions and the dates of receipt of correspondence filed in the Office, may also be accessed via the Internet through the Patent Electronic Business Center at www.uspto.gov using the public side of the Patent Application Information and Retrieval (PAIR) system. The direct link to access this status information is currently http://pair.uspto.gov/. Prior to publication, such status information is confidential and may only be obtained by applicant using the private side of PAIR.

Further assistance in electronically accessing the publication, or about PAIR, is available by calling the Patent Electronic Business Center at 703-305-3028.

Pre-Grant Publication Division, 703-605-4283

JAN 18 2007
PATENT & TRADEMARK OFFICE

Docket Number: 011765-0355986
Client Reference: AF/JG/P10717US

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re the Application of

MELVYN MORRIS, et al.

Group Art Unit: 2131

Application No.: 11/477,807

Examiner: Unassigned

Filed: June 30, 2006

Confirmation No.: 4785

For: METHODS AND APPARATUS FOR DEALING WITH MALWARE

INFORMATION DISCLOSURE STATEMENT

Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

Sir:

Pursuant to 37 CFR 1.56, the attention of the Patent and Trademark Office is hereby directed to the references listed on the attached PTO-1449. Copies of the required references are provided herewith. It is respectfully requested that the information be expressly considered during the prosecution of this application, and that the references be made of record therein and appear among the "References Cited" on any patent to issue therefrom. Applicants respectfully request the Examiner return an initialed copy of the enclosed Form PTO-1449 to Applicants with the next Office communication to indicate that the references have been considered, per MPEP § 609.

This Information Disclosure Statement is being filed before the mailing of the first Office Action on the merits, and therefore no fee is required herewith.

Respectfully Submitted,
PILLSBURY WINTHROP SHAW PITTMAN LLP

Christine H. McCarthy
Registration Number 41844
Customer Number: 00909

Date: January 18, 2007
Telephone: (703) 770-7900
Facsimile: (703) 770-7901
P.O. Box 10500
McLean, VA 22102

OIPE IAP97
JAN 18 2007
PATENT & TRADEMARK OFFICE

FORM PTO-1449 (modified)
To: U.S. Department of Commerce
(PW FORM PAT-1449) Patent and Trademark Office

Atty. Dkt. No.	M#	Client Ref.
	11765-0355986	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

Applicant: MORRIS et al.

Appln. No.: 11/477,807

Filing Date: June 30, 2006

Date: January 18, 2007 Page 1 of 1

Examiner: Unassigned | Group Art Unit: 2131

U.S. PATENT DOCUMENTS

Examiner's Initials*		Document Number	Date MM/YYYY	Name (Family Name of First Inventor)	Class	Sub Class	Filing Date (if appropriate)
	AR	6,772,346	08/2004	CHESS			
	BR	6,944,772	09/2005	DOZORTSEV			
	CR	2004/0006704	01/2004	DAHLSTROM			
	DR	2004/0073810	05/2004	RACINE			
	ER	2005/0021994	01/2005	BARTON			
	FR	2004/0153644	08/2004	McCORKENDALE			
	GR						
	HR						
	IR						
	JR						
	KR						
	LR						
	MR						
	NR						

FOREIGN PATENT DOCUMENTS

		Document Number	Date MM/YYYY	Country	Inventor Name		English Abstract		Translation Readily Available	
							Enclosed	No	Enclose	No
	OR	1 280 040	01/2003	EP						
	PR	1 549 012	12/2003	EP						
	QR	03/021402	03/2003	WO						
	RR	2004/097602	11/2004	WO						
	SR	02/33525	04/2002	WO						

OTHER (Including in this order Author, Title, Periodical Name, Date, Pertinent Pages, etc.)

	TR				
	UR				
	VR				
	WR				

Examiner | Date Considered:

*EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP § 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to Applicant.

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 280 040 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
29.01.2003 Bulletin 2003/05

(51) Int Cl.[7]: **G06F 1/00**

(21) Application number: **02253235.2**

(22) Date of filing: **09.05.2002**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **26.07.2001 US 912392**

(71) Applicant: **Networks Associates Technology Inc.**
Santa Clara, CA 95054 (US)

(72) Inventors:
- **Hinchliffe, Alexander James**
  Milton Keynes, Buckinghamshire MK7 7LL (GB)
- **Howard, Fraser Peter**
  Chipping Norton, Oxon OX7 3BT (GB)
- **Kemp, Andrew**
  Oxford, Oxon OX4 2QT (GB)
- **Rai, Bobby**
  Kettering, Northants NN15 5BP (GB)

(74) Representative:
Robinson, Nigel Alexander Julian
D. Young & Co.,
21 New Fetter Lane
London EC4A 1DA (GB)

(54) Centrally managed virus scanning system

(57) A malware detecting system is described in which a client computer 8 sends a file access clearance request to an assessment computer 24 to determine whether or not access is permitted to that file. In this way, the different client computers on a network 2 may share their scan results and centralised control may be provided over access permissions to individual files or groups of files.

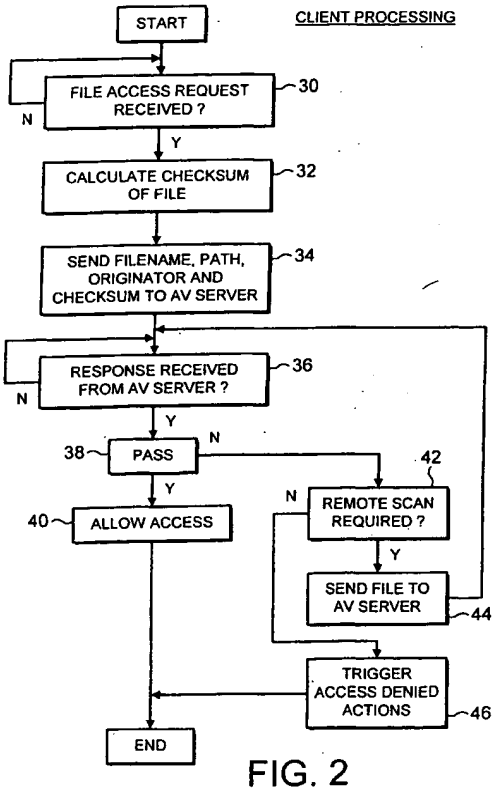


FIG. 2

EP 1 280 040 A2

Trend Micro, Inc.
EX1007-537

Description

[0001] This invention relates to the field of data processing systems. More particularly, this invention relates to the field of malware scanning, such as, for example, scanning for computer viruses, Trojans, worms, banned computer files or E-mails containing banned words or content.

[0002] It is known to provide malware scanning systems and mechanisms for identifying malware within a computer file to be accessed. As the number of computer viruses and the like that are present in the wild increases the processing associated with scanning a computer file to identify the presence of any of those viruses similarly increases. This increase in the processing required is disadvantageous. Furthermore, with the increasing levels of network connectivity between computer systems and the use of E-mail and other fast messaging systems, the spread of computer viruses has become increasingly rapid. In a number of recent cases mass mailing viruses have spread at such a speed that considerable damage has been caused before appropriate countermeasures have been able to be put in place. The delay in deploying such countermeasures is further increased by the need in many cases to update virus definition data on every individual node computer to be protected. The speed of spread of recent viruses is such that the download and installation delays associated with installing such new countermeasures, even when they have been developed and are available, is a significant disadvantage.

[0003] An anti-virus system produced by Sophos uses a locally held record of previously conducted on-access scans on the individual computer in question to determine whether or not a scan should be conducted for a computer file or a previously determined result used instead when this is available.

[0004] Viewed from one aspect the present invention provides a computer program product for controlling a computer to detect malware, said computer program product comprising:

detecting logic operable to detect a file access request to a computer file by a requesting computer;
file access clearance request generating logic operable to generate a file access clearance request including data identifying said computer file;
file access clearance request transmitting logic operable to transmit said file access clearance request from said requesting computer to an assessment computer responsible for assessment of whether said computer file contains malware;
file access clearance response receiving logic operable to receive at said requesting computer a file access clearance response from said assessment computer; and
file access permitting logic operable if said file access clearance response indicates said computer file does not contain malware, to permit said file access request by said requesting computer.

[0005] The invention recognises that an individual computer requiring a file access may pass the task of determining whether or not that access should be allowed to a different assessment computer. Whilst at first sight this may seem that it would slow down the file access, in practice there are considerable advantages. For example, the assessment computer to which the task is passed may be more rapidly updated with new virus definition data as this is released than would be possible for the requesting computer. Accordingly, the requesting computer can benefit from the most up-to-date virus definition data more rapidly. Furthermore, the invention is particularly well suited to systems in which a plurality of requesting computers share an assessment computer since in many cases the individual requesting computers will show a high degree of correlation in the computer files to which they are requesting access and for which a malware scan is needed. Thus, rather than each requesting computer individually scanning the computer files that are also being scanned by a large number of other computers, (e.g. the computer files associated with the operating system shared by all the requesting computers), these computer files can instead be scanned once by the assessment computer and then each requesting computer can check that the computer file has been scanned and cleared by sending an appropriate request to the assessment computer rather than having to scan the file itself. Thus, in exchange for the use of a small amount of bandwidth on the connections between the requesting computers and the assessment computer, a considerable processing burden may be lifted from the requesting computers. The additional processing burden on the assessment computer does not increase disproportionately since the high degree of correlation between the computer files accessed by the different requesting computers means that in many cases the assessment computer will be able to respond to a clearance request from a requesting computer on the basis that it has already scanned that file and without the need to rescan that file.

[0006] In order to identify the computer file to be scanned in a secure manner to the assessment computer, the data identifying the computer file preferably includes a checksum value. Checksum values may be made sufficiently specific for a computer file from which they have been calculated so as to be difficult to bypass as a way of uniquely identifying a computer file.

[0007] Additional data that is highly useful to the assessment computer in managing the requests it receives includes the filename of the computer file, data identifying the requesting computer and the storage location of the computer file. This provides useful audit information as well as providing for the possibility of files having the same name and same storage location on different computers in fact being different files with dif-

Trend Micro, Inc.
EX1007-538

ferent checksums, both of which should be treated as separate entities when determining whether or not a request from another requesting computer relates to a file that has already scanned.

[0008] In circumstances when a requesting computer wishes to access a file that has not already been scanned, the assessment computer may send a scan request message back to the requesting computer such that the requesting computer may send a copy of the file to the assessment computer for scanning. It will be appreciated that this may be slower than scanning the file locally, but the benefit will be that should a different requesting computer later wish to access that same file, then the scanning result produced by this action can be shared with that other computer, so speeding the operation of that other computer.

[0009] If the assessment computer determines that access should be denied to a computer file, then this may be used to trigger denied access actions within the requesting computer, the assessment computer, or elsewhere. Such denied access actions may include deletion of the computer file, repair of the computer file, quarantining of the computer file, generation of user warning messages, generation of administrator warning messages and the like.

[0010] Viewed from another aspect the invention also provides a computer program product for controlling a computer to detect malware, said computer program product comprising:

file access request receiving logic operable to receive at an assessment computer a file access clearance request from a requesting computer, said file access clearance request including data identifying a computer file to be accessed by said requesting computer;

file access clearance response generating logic operable in dependence upon said data identifying said computer file to determine if said computer file has previously been assessed as not containing malware and to generate a file access clearance response; and

file access clearance response transmitting logic operable to transmit said file access clearance response to said requesting computer.

[0011] The assessment computer may advantageously store the results of its previous scans within a database. This database may include a variety of fields relating to each computer file that has been scanned. These fields may include an access flag indicating whether access is to be denied to that computer file and a persistence flag indicating whether or not the entry for that computer file should be purged during purge operations.

[0012] The access flag is particularly useful as in addition to allowing the recording that access should be denied due to malware being detected, it also allows the central management of whether or not a particular individual file or class or type of file should be permitted to be accessed by all of those requesting computers that seek their access permissions from that assessment computer. This centralised control is a powerful tool that may be used to implement techniques such as the triggering of a lock down mode of operation in which higher level security provisions are put in place by denying access to certain files or types of files. As an example, if a message was received indicating that the higher security mode should be entered, then the assessment computer may use its mechanisms to deny access to any newly encountered computer file that had not previously been scanned and cleared for use. This would typically allow the large majority of computer activity to continue whilst providing protection against newly released malware threats until the appropriate countermeasures could be put in place.

[0013] The persistence flag allows control of the flushing of entries from the assessment computer. Whilst one of the advantages of the invention is storing the results of previously conducted scans such that they need not be repeated, this has to be tempered by allowing the results to be refreshed at a later time for at least some files. It is possible to envisage that a particular computer file carrying a newly released virus may not be detected as carrying that virus when it is first scanned, but later when a new virus driver is available, that computer file would be detected and blocked. Accordingly, as an example, it may be that all previous scan results could be purged from the system whenever the virus definition data was updated.

[0014] Viewed from a further aspect the invention provides a computer program product for controlling a computer to detect malware, said computer program product comprising:

file access request detecting logic operable to detect a file access request to a computer file by a requesting computer;

file access clearance request generating logic operable to generate a file access clearance request including data identifying said computer file;

file access clearance request transmitting logic operable to transmit said file access clearance request from said requesting computer to an assessment computer responsible for assessment of whether said computer file contains malware;

file access clearance request receiving logic operable to receive at said assessment computer said file access clearance request from a requesting computer;

file access clearance response generating logic operable in dependence upon said data identifying said computer file to determine if said computer file has previously been assessed as not containing malware and to generate a file access clearance response;

Trend Micro, Inc.
EX1007-539

file access clearance response transmitting logic operable to transmit said file access clearance response to said requesting computer;
file access clearance response receiving logic operable to receive at said requesting computer said file access clearance response from said assessment computer; and
file access permitting logic operable if said file access clearance response indicates said computer file does not contain malware to permit said file access request by said requesting computer.

[0015] As well as the complementary aspects of the invention embodied in the form of the client software, the server software and the combination of the client and server software, the invention may also take the form of corresponding methods of malware detection and apparatus for malware detection.

[0016] Embodiments of the invention will now be described, by way of example only, with reference to the accompanying drawings in which:

Figure 1 schematically illustrates a computer network to which the present technique may be applied;

Figure 2 is a flow diagram illustrating an example of the processing that may be performed on a client computer;

Figure 3 is a flow diagram illustrating an example of the processing that may be performed on a server computer;

Figure 4 is a flow diagram illustrating an example process that may be run on a server computer waiting for a lock down signal;

Figure 5 schematically illustrates a database entry relating to a computer file that has previously been scanned; and

Figure 6 is a diagram schematically illustrating the form of a general purpose computer that may be used to implement the above described techniques.

[0017] Figure 1 schematically illustrates a computer network 2 comprising a plurality of client computers 4, 6, 8, 10, 12, 14, 16, 18 connected to respective departmental servers 20, 22 and to a corporate anti-virus (assessment) server 24. The computer network 2 is also connected via a network link to an anti-virus providers FTP server 26 from which virus definition data may be downloaded and from which messages, such as lock down messages or messages relating to the availability of new virus definition data may be received.

[0018] In operation a client computer 8 conducts processing activity during which it seeks to make a file access to a particular computer file (in this example, winword.exe). Rather than scanning this file itself for malware such as computer viruses, Trojans, worms, banned content etc, the client computer 8 instead passes an access clearance request relating to the computer file through to the assessment computer 24 via the departmental server 20. The access clearance request includes the file name of the computer file being accessed, a checksum derived from that file in an effort to uniquely identify it (e.g. an MD5 checksum), data identifying the name of the client computer 8, and the path used by that client computer 8 to access the computer file. The assessment computer 24 receives the access clearance request and uses the data identifying the computer file to look up within an associated database 28 as to whether or not a malware scan has already been conducted for that particular computer file. The computer file is uniquely identified by its filename and checksum value. If the computer file in question has already been scanned, then the result of that scan may be reused rather than conducting the scan again. The assessment computer 24 can accordingly pass back to the client computer 8 a clearance request response indicating whether or not access to that file is permitted or some other further action should be taken. The client computer 8 can then use this access clearance response to either permit access to the file or take the further action specified.

[0019] The further action specified could be that the computer file has already been identified as containing malware and accordingly an appropriate anti-virus or other anti-malware response should be initiated on the client computer 8. A corresponding response could already have been initiated on the assessment computer 24. The denied access responses could take the form of deleting the computer file concerned, repairing the computer file concerned, quarantining the computer file concerned, issuing a user alerting message, issuing an administrator alerting message, issuing a message to an anti-virus provider or some other action.

[0020] A different type of further action that could be required by the client computer 8 in response to the access clearance response arises when the assessment computer 24 has not previously scanned that file and accordingly the client computer 8 should send a copy of that file through to the assessment computer 24 to be scanned. When such a copy of the computer file has been sent through to the assessment computer 24 and scanned, an appropriate pass or denied access action triggering response can be sent back from the assessment computer 24 to the client computer 8.

[0021] It will be appreciated that the client computers 4, 6, 8, 10, 12, 14, 16, 18 operate to share the results of the malware scanning that is performed on their behalf by the assessment computer 24. Since there is likely to be a high degree of correlation between the files being accessed by the different client computers, in many cases the scan that a client computer is request-

Trend Micro, Inc.
EX1007-540

ing will already have been performed on behalf of another computer and that scan need not be run again by the assessment computer but instead merely the result of the previous scan returned. The use of a checksum allows the computer files to be reliably uniquely identified.

[0022] The co-ordination of file access permissions by the assessment computer 24 also allows central management of which computer files may be accessed upon the network 2 in a manner that allows the rapid implementation of any higher security level mode should this be desired. As an example, a lock down trigger message may be received from the anti-virus provider FTP server 26 that will trigger the assessment computer 24 to enter a higher security level mode compared to its normal mode of operation. In the higher security level mode, it may be that the access to whole variety of different types of files may be temporarily banned. As an example, the access to VBS files which are often a source of viruses may be banned across the network 2, as could access to e-mail attachments which are another potential source of virus propagation. Thus, when a new threat is identified by the anti-virus provider, lock down messages may be used to trigger predetermined, in accordance with user configuration and preferences, higher security level modes within connected assessment computers 24 to provide a degree of protection for the networks concerned whilst allowing the majority of their normal operations to continue.

[0023] Figure 2 is a flow diagram illustrating the processing performed by a client computer. At step 30 the computer waits to receive a file access request from a computer program executing on that client computer. At step 32 a checksum value is calculated in accordance with one of several different possible checksum algorithms, such as the MD5 algorithm. At step 34 the filename, path, originating computer name and checksum value are sent through to the assessment computer. At step 36 the client computer waits to receive a response from the assessment computer. When that response is received, step 38 serves to determine whether or not the response indicates that the computer file had passed its scanning. If the computer file had passed its scanning and access is to be permitted, then processing proceeds to step 40 at which the access is allowed.

[0024] If the response is not that the computer file had passed its scan, then processing proceeds to step 42 at which a test is made as to whether or not the response indicated that a remote scan was required. If a remote scan was required, then the computer file in question is sent to the assessment computer at step 44 and processing is returned to step 36.

[0025] If the test at step 42 does not identify a remote scan request, then since the response is neither a pass, or a remote scan request, then the computer file must be one to which access is denied and accordingly step 46 serves to trigger the denied access actions. These may include deletion, repair, quarantining or other actions upon the computer file in question as well as the generation of appropriate warning messages to a user or an administrator.

[0026] Figure 3 illustrates the processing performed by the assessment computer. At step 48 the assessment computer waits to receive a request from a client computer. When a request is received, step 50 serves to identify whether a copy of a computer file is being returned to the assessment computer for scanning following a remote scan request that had earlier been issued by the assessment computer. If a copy of a computer file is being returned for scanning, then processing proceeds to step 52 at which the necessary malware scanning is conducted. Step 54 determines whether or not the computer file passed this malware scanning. If the computer file did pass the malware scanning, then step 56 serves to add details of that computer file to the database of scanned files held at the assessment computer relating to scans already performed. Step 58 then returns the pass result to the client computer.

[0027] If the scanning at step 52 was detected as not being passed at step 54, then processing proceeds to step 60 at which malware detected actions are triggered within the assessment computer. These malware detected actions may be similar to those previously described in relation to the client computer file. In addition, an entry specifying the malware scan fail of that computer file may also be added to the database. At step 62 an access denied result is returned to the client computer.

[0028] If the test at step 50 did not indicate that a computer file was being returned for scan, then step 64 serves to compare the file details being passed to the server with the database entries of computer files that have already been scanned. The computer file may be uniquely identified by its filename and its checksum. If at step 66 a match within the database is detected, then step 68 determines whether or not this entry indicates that access should be allowed to that computer file. If access is not to be allowed, then processing proceeds to step 60. If access is allowed, then processing proceeds to step 70 at which a pass result is returned to the client computer.

[0029] If the test at step 66 indicated that a match was not found within the database, then step 72 serves to return a remote scan required result to the client computer in order to trigger the client computer to return a copy of the computer file to the assessment computer for scanning at step 52.

[0030] Figure 4 illustrates a process that may run on the assessment computer as a background task. At step 74 the assessment computer waits to receive a lock down trigger message from the anti-virus provider. If such a message is received, then step 76 serves to activate a lock down mode in the assessment computer. The lock down mode can switch on a user predetermined set of measures intended to provide a higher degree of security, normally at the cost of at least some

Trend Micro, Inc.
EX1007-541

functionality. As an example, access to e-mail attachments or VBS files as a class may be denied. These are known to be particular vulnerabilities.

[0031] At step 78 the assessment computer waits for a user command to cancel the lock down mode. When such a command is received, then step 80 serves to cancel the lock down mode.

[0032] Figure 5 schematically illustrates the data that may be stored for a particular computer file within the database of previously conducted scan results held by the assessment computer 24. For an individual computer file, its filename, the originating computer for which the scan of that file was first conducted, the path to the file, the checksum value for the file, an allowed access flag and a persistence flag are all stored. The allowed access flag may be used to indicate whether or not that file passed its scan result. The allowed access flag may also be used as a powerful tool for switching off or on access to individual files or classes of files by an administrator. The persistence flag controls how the entry is flushed from the database on a regular interval, such as when new virus definition data is received. It will be appreciated that other fields could be added to the database relating to the particular file as required.

[0033] Figure 6 schematically illustrates a general purpose computer 200 of the type that may be used to implement the above techniques. The general purpose computer 200 includes a central processing unit 202, a random access memory 204, a read only memory 206, a hard disk drive 208, a display driver 210 and display 212, a user input/output circuit 214 and keyboard 216 and mouse 218 and a network interface unit 220 all connected via a common bus 222. In operation the central processing unit 202 executes program instructions stored within the random access memory 204, the read only memory 206 or the hard disK drive 208. The working memory is provided by the random access memory 204. The program instructions could take a variety of forms depending on the precise nature of the computer 200 and the programming language being used. The results of the processing are displayed to a user upon the display 212 driven by the display driver 210. User inputs for controlling the general purpose computer 200 are received from the keyboard 216 and the mouse 218 via the user input/output circuit 214. Communication with other computers, such as exchanging e-mails, downloading files or providing internet or other network access, is achieved via the network interface unit 220.

[0034] It will be appreciated that the general purpose computer 200 operating under control of a suitable computer program may perform the above described techniques and provide apparatus for performing the various tasks described. The general purpose computer 200 also executes the method described previously. The computer program product could take the form of a recordable medium bearing the computer program, such as a floppy disk, a compact disk or other recordable medium. Alternatively, the computer program could be dynamically downloaded via the network interface unit 220.

[0035] It will be appreciated that the general purpose computer 200 is only one example of the type of computer architecture that may be employed to carry out the above described techniques. Alternative architectures are envisaged and are capable of use with the above described techniques.

Claims

1. A computer program product for controlling a computer to detect malware, said computer program product comprising:

detecting logic operable to detect a file access request to a computer file by a requesting computer;
file access clearance request generating logic operable to generate a file access clearance request including data identifying said computer file;
file access clearance request transmitting logic operable to transmit said file access clearance request from said requesting computer to an assessment computer responsible for assessment of whether said computer file contains malware;
file access clearance response receiving logic operable to receive at said requesting computer a file access clearance response from said assessment computer; and
file access permitting logic operable if said file access clearance response indicates said computer file does not contain malware, to permit said file access request by said requesting computer.

2. A computer program product as claimed in claim 1, wherein said data identifying said computer file includes a checksum value calculated from said computer file.

3. A computer program product as claimed in claims 1 and 2, wherein said data identifying said computer file includes one or more of a filename of said computer file, data identifying said requesting computer and a storage location of said computer file.

4. A computer program product as claimed in claims 1, 2 and 3, wherein if said file access clearance response indicates a scan of said computer file is required by said assessment computer, then computer file transmitting logic is operable to transmit said computer file from said requesting computer to said assessment computer.

5. A computer program product as claimed in any

Trend Micro, Inc.
EX1007-542

one of the preceding claims, wherein if said file access clearance response indicates access to said computer file is denied, then triggering a denied access response in said requesting computer.

6. A computer program product as claimed in any one of the preceding claims, wherein a plurality of requesting computers share access to an assessment computer for determining whether file access requests by those requesting computers should be denied.

7. A computer program product for controlling a computer to detect malware, said computer program product comprising:

file access request receiving logic operable to receive at an assessment computer a file access clearance request from a requesting computer, said file access clearance request including data identifying a computer file to be accessed by said requesting computer;
file access clearance response generating logic operable in dependence upon said data identifying said computer file to determine if said computer file has previously been assessed as not containing malware and to generate a file access clearance response; and
file access clearance response transmitting logic operable to transmit said file access clearance response to said requesting computer.

8. A computer program product as claimed in claim 7, wherein said data identifying said computer file includes a checksum value calculated from said computer file.

9. A computer program product as claimed in any one of claims 7 and 8, wherein said data identifying said computer file includes one or more of a filename of said computer file, data identifying said requesting computer and a storage location of said computer file.

10. A computer program product as claimed in any one of claim 7, 8 and 9, wherein if said file access clearance response indicates a scan of said computer file is required by said assessment computer, then computer file receiving logic is operable to receive at said assessment computer said computer file from said requesting computer and performing a malware scan of said computer file.

11. A computer program product as claimed in any one of claims 7 to 10, wherein if said file access clearance response indicates access to said computer file is denied, then triggering a denied access response in said assessment computer.

12. A computer program product as claimed in any one of claims 7 to 11, wherein said assessment computer stores a database of computer files previously assessed as to whether they contain malware.

13. A computer program product as claimed in claim 12, wherein said database includes for each computer file fields specifying one or more of a filename of said computer file, data identifying said requesting computer and a storage location of said computer file, a checksum value calculated from said computer file, an access flag indicating whether access to said computer file is denied and a persistence flag indicating whether entries relating to said computer file should be purged from said database during purge operations.

14. A computer program product as claimed in any one of claims 7 to 13, wherein said assessment computer is operable in at least a higher level security mode and a lower level security mode, said assessment computer serving to deny access to greater range of computer files when operating in said higher level security mode compared with said lower level security mode.

15. A computer program product as claimed in claim 14, wherein said assessment computer is triggered to change from said lower level security mode to said higher level security mode by a lock down trigger message received at said assessment computer from a remote computer.

16. A computer program product as claimed in any one of claims 7 to 15, wherein a plurality of requesting computers share access to an assessment computer for determining whether file access requests by those requesting computers should be denied.

17. A computer program product for controlling a computer to detect malware, said computer program product comprising:

file access request detecting logic operable to detect a file access request to a computer file by a requesting computer;
file access clearance request generating logic operable to generate a file access clearance request including data identifying said computer file;
file access clearance request transmitting logic operable to transmit said file access clearance request from said requesting computer to an assessment computer responsible for assessment of whether said computer file contains malware;
file access clearance request receiving logic

Trend Micro, Inc.
EX1007-543

operable to receive at said assessment computer said file access clearance request from a requesting computer;
file access clearance response generating logic operable in dependence upon said data identifying said computer file to determine if said computer file has previously been assessed as not containing malware and to generate a file access clearance response;
file access clearance response transmitting logic operable to transmit said file access clearance response to said requesting computer;
file access clearance response receiving logic operable to receive at said requesting computer said file access clearance response from said assessment computer; and
file access permitting logic operable if said file access clearance response indicates said computer file does not contain malware to permit said file access request by said requesting computer.

18. A computer program product as claimed in claim 17, wherein said data identifying said computer file includes a checksum value calculated from said computer file.

19. A computer program product as claimed in claims 17 and 18, wherein said data identifying said computer file includes one or more of a filename of said computer file, data identifying said requesting computer and a storage location of said computer file.

20. A computer program product as claimed in any one of claims 17 to 19, wherein if said file access clearance response indicates a scan of said computer file is required by said assessment computer, then computer file transmitting logic is operable to transmit said computer file from said requesting computer to said assessment computer, receiving at said assessment computer said computer file from said requesting computer and performing a malware scan of said computer file.

21. A computer program product as claimed in any one of claims 17 to 20, wherein if said file access clearance response indicates access to said computer file is denied, then triggering a denied access response in said assessment computer.

22. A computer program product as claimed in any one of claims 17 to 21, wherein if said file access clearance response indicates access to said computer file is denied, then triggering a denied access response in said requesting computer.

23. A computer program product as claimed in any one of claims 17 to 22, wherein said assessment computer stores a database of computer files previously assessed as to whether they contain malware.

24. A computer program product as claimed in claim 23, wherein said database includes for each computer file fields specifying one or more of a filename of said computer file, data identifying said requesting computer and a storage location of said computer file, a checksum value calculated from said computer file, an access flag indicating whether access to said computer file is denied and a persistence flag indicating whether entries relating to said computer file should be purged from said database during purge operations.

25. A computer program product as claimed in any one of claims 17 to 25, wherein said assessment computer is operable in at least a higher level security mode and a lower level security mode, said assessment computer serving to deny access to greater range of computer files when operating in said higher level security mode compared with said lower level security mode.

26. A computer program product as claimed in claim 25, wherein said assessment computer is triggered to change from said lower level security mode to said higher level security mode by a lock down trigger message received at said assessment computer from a remote computer.

28. A computer program product as claimed in any one of claims 17 to 27, wherein a plurality of requesting computers share access to an assessment computer for determining whether file access requests by those requesting computers should be denied.

29. A method of detecting malware, said method comprising the steps of:

detecting a file access request to a computer file by a requesting computer;
generating a file access clearance request including data identifying said computer file;
transmitting said file access clearance request from said requesting computer to an assessment computer responsible for assessment of whether said computer file contains malware;
receiving at said requesting computer a file access clearance response from said assessment computer; and
if said file access clearance response indicates said computer file does not contain malware, then permitting said file access request by said requesting computer.

Trend Micro, Inc.
EX1007-544

30. A method as claimed in claim 29, wherein said data identifying said computer file includes a checksum value calculated from said computer file.

31. A method as claimed in claims 29 and 30, wherein said data identifying said computer file includes one or more of a filename of said computer file, data identifying said requesting computer and a storage location of said computer file.

32. A method as claimed in any one of claims 29 to 31, wherein if said file access clearance response indicates a scan of said computer file is required by said assessment computer, then transmitting said computer file from said requesting computer to said assessment computer.

33. A method as claimed in any one of claims 29 to 31, wherein if said file access clearance response indicates access to said computer file is denied, then triggering a denied access response in said requesting computer.

34. A method as claimed in any one of claims 29 to 33, wherein a plurality of requesting computers share access to an assessment computer for determining whether file access requests by those requesting computers should be denied.

35. A method of detecting malware, said method comprising the steps of:

receiving at an assessment computer a file access clearance request from a requesting computer, said file access clearance request including data identifying a computer file to be accessed by said requesting computer;
in dependence upon said data identifying said computer file determining if said computer file has previously been assessed as not containing malware and generating a file access clearance response; and
transmitting said file access clearance response to said requesting computer.

36. A method as claimed in claim 35, wherein said data identifying said computer file includes a checksum value calculated from said computer file.

37. A method as claimed in claims 35 and 36, wherein said data identifying said computer file includes one or more of a filename of said computer file, data identifying said requesting computer and a storage location of said computer file.

38. A method as claimed in any one of claims 35 to 37, wherein if said file access clearance response indicates a scan of said computer file is required by said assessment computer, then receiving at said assessment computer said computer file from said requesting computer and performing a malware scan of said computer file.

39. A method as claimed in any one of claims 35 to 38, wherein if said file access clearance response indicates access to said computer file is denied, then triggering a denied access response in said assessment computer.

40. A method as claimed in any one of claims 35 to 39, wherein said assessment computer stores a database of computer files previously assessed as to whether they contain malware.

41. A method as claimed in claim 40, wherein said database includes for each computer file fields specifying one or more of a filename of said computer file, data identifying said requesting computer and a storage location of said computer file, a checksum value calculated from said computer file, an access flag indicating whether access to said computer file is denied and a persistence flag indicating whether entries relating to said computer file should be purged from said database during purge operations.

42. A method as claimed in any one of claims 35 to 41, wherein said assessment computer is operable in at least a higher level security mode and a lower level security mode, said assessment computer serving to deny access to greater range of computer files when operating in said higher level security mode compared with said lower level security mode.

43. A method as claimed in any one of claims 35 to 42, wherein said assessment computer is triggered to change from said lower level security mode to said higher level security mode by a lock down trigger message received at said assessment computer from a remote computer.

44. A method as claimed in any one of claims 35 to 43, wherein a plurality of requesting computers share access to an assessment computer for determining whether file access requests by those requesting computers should be denied.

45. A method of detecting malware, said method comprising the steps of:

detecting a file access request to a computer file by a requesting computer;
generating a file access clearance request including data identifying said computer file;
transmitting said file access clearance request

Trend Micro, Inc.
EX1007-545

from said requesting computer to an assessment computer responsible for assessment of whether said computer file contains malware;
receiving at said assessment computer said file access clearance request from a requesting computer;
in dependence upon said data identifying said computer file determining if said computer file has previously been assessed as not containing malware and generating a file access clearance response;
transmitting said file access clearance response to said requesting computer;
receiving at said requesting computer said file access clearance response from said assessment computer; and
if said file access clearance response indicates said computer file does not contain malware, then permitting said file access request by said requesting computer.

46. A method as claimed in claim 45, wherein said data identifying said computer file includes a checksum value calculated from said computer file.

47. A method as claimed in claims 45 and 46, wherein said data identifying said computer file includes one or more of a filename of said computer file, data identifying said requesting computer and a storage location of said computer file.

48. A method as claimed in any one of claims 45 to 47, wherein if said file access clearance response indicates a scan of said computer file is required by said assessment computer, then transmitting said computer file from said requesting computer to said assessment computer, receiving at said assessment computer said computer file from said requesting computer and performing a malware scan of said computer file.

49. A method as claimed in any one of claims 45 to 48, wherein if said file access clearance response indicates access to said computer file is denied, then triggering a denied access response in said assessment computer.

50. A method as claimed in any one of claims 45 to 49, wherein if said file access clearance response indicates access to said computer file is denied, then triggering a denied access response in said requesting computer.

51. A method as claimed in any one of claims 45 to 50, wherein said assessment computer stores a database of computer files previously assessed as to whether they contain malware.

52. A method as claimed in claim 51, wherein said database includes for each computer file fields specifying one or more of a filename of said computer file, data identifying said requesting computer and a storage location of said computer file, a checksum value calculated from said computer file, an access flag indicating whether access to said computer file is denied and a persistence flag indicating whether entries relating to said computer file should be purged from said database during purge operations.

53. A method as claimed in any one of claims 45 to 52, wherein said assessment computer is operable in at least a higher level security mode and a lower level security mode, said assessment computer serving to deny access to greater range of computer files when operating in said higher level security mode compared with said lower level security mode.

54. A method as claimed in claim 53, wherein said assessment computer is triggered to change from said lower level security mode to said higher level security mode by a lock down trigger message received at said assessment computer from a remote computer.

55. A method as claimed in any one of claims 45 to 54, wherein a plurality of requesting computers share access to an assessment computer for determining whether file access requests by those requesting computers should be denied.

56. Apparatus for detecting malware, said apparatus comprising:

a detector operable to detect a file access request to a computer file by a requesting computer;
a file access clearance request generator operable to generate a file access clearance request including data identifying said computer file;
a file access clearance request transmitter operable to transmit said file access clearance request from said requesting computer to an assessment computer responsible for assessment of whether said computer file contains malware;
a file access clearance response receiver operable to receive at said requesting computer a file access clearance response from said assessment computer; and
a file access permission unit operable if said file access clearance response indicates said computer file does not contain malware, to permit said file access request by said requesting

Trend Micro, Inc.
EX1007-546

computer.

57. Apparatus as claimed in claim 56, wherein said data identifying said computer file includes a checksum value calculated from said computer file.

58. Apparatus as claimed in claims 56 and 57, wherein said data identifying said computer file includes one or more of a filename of said computer file, data identifying said requesting computer and a storage location of said computer file.

59. Apparatus as claimed in any one of claims 56 to 58, wherein if said file access clearance response indicates a scan of said computer file is required by said assessment computer, then a computer file transmitter is operable to transmit said computer file from said requesting computer to said assessment computer.

60. Apparatus as claimed in any one of claims 56 to 59, wherein if said file access clearance response indicates access to said computer file is denied, then triggering a denied access response in said requesting computer.

61. Apparatus as claimed in any one of claims 56 to 60, wherein a plurality of requesting computers share access to an assessment computer for determining whether file access requests by those requesting computers should be denied.

62. Apparatus for controlling a computer to detect malware, said apparatus comprising:

a file access request receiver operable to receive at an assessment computer a file access clearance request from a requesting computer, said file access clearance request including data identifying a computer file to be accessed by said requesting computer;
a file access clearance response generator operable in dependence upon said data identifying said computer file to determine if said computer file has previously been assessed as not containing malware and to generate a file access clearance response; and
a file access clearance response transmitter operable to transmit said file access clearance response to said requesting computer.

63. Apparatus as claimed in claim 62, wherein said data identifying said computer file includes a checksum value calculated from said computer file.

64. Apparatus as claimed in claims 62 and 63, wherein said data identifying said computer file includes one or more of a filename of said computer file, data identifying said requesting computer and a storage location of said computer file.

65. Apparatus as claimed in any one of claims 62 to 64, wherein if said file access clearance response indicates a scan of said computer file is required by said assessment computer, then a computer file receiver is operable to receive at said assessment computer said computer file from said requesting computer and performing a malware scan of said computer file.

66. Apparatus as claimed in any one of claims 62 to 65, wherein if said file access clearance response indicates access to said computer file is denied, then triggering a denied access response in said assessment computer.

67. Apparatus as claimed in any one of claims 62 to 66, wherein said assessment computer stores a database of computer files previously assessed as to whether they contain malware.

68. Apparatus as claimed in claim 67, wherein said database includes for each computer file fields specifying one or more of a filename of said computer file, data identifying said requesting computer and a storage location of said computer file, a checksum value calculated from said computer file, an access flag indicating whether access to said computer file is denied and a persistence flag indicating whether entries relating to said computer file should be purged from said database during purge operations.

69. Apparatus as claimed in any one of claims 62 to 68, wherein said assessment computer is operable in at least a higher level security mode and a lower level security mode, said assessment computer serving to deny access to greater range of computer files when operating in said higher level security mode compared with said lower level security mode.

70. Apparatus as claimed in claim 69, wherein said assessment computer is triggered to change from said lower level security mode to said higher level security mode by a lock down trigger message received at said assessment computer from a remote computer.

71. Apparatus as claimed in any one of claims 62 to 70, wherein a plurality of requesting computers share access to an assessment computer for determining whether file access requests by those requesting computers should be denied.

72. Apparatus for controlling a computer to detect

Trend Micro, Inc.
EX1007-547

malware, said apparatus comprising:

a file access request detector operable to detect a file access request to a computer file by a requesting computer;
a file access clearance request generator operable to generate a file access clearance request including data identifying said computer file;
a file access clearance request transmitter operable to transmit said file access clearance request from said requesting computer to an assessment computer responsible for assessment of whether said computer file contains malware;
a file access clearance request receiver operable to receive at said assessment computer said file access clearance request from a requesting computer;
a file access clearance response generator operable in dependence upon said data identifying said computer file to determine if said computer file has previously been assessed as not containing malware and to generate a file access clearance response;
a file access clearance response transmitter operable to transmit said file access clearance response to said requesting computer;
a file access clearance response receiver operable to receive at said requesting computer said file access clearance response from said assessment computer; and
a file access permission unit operable if said file access clearance response indicates said computer file does not contain malware to permit said file access request by said requesting computer.

73. Apparatus as claimed in claim 72, wherein said data identifying said computer file includes a checksum value calculated from said computer file.

74. Apparatus as claimed in claims 72 and 73, wherein said data identifying said computer file includes one or more of a filename of said computer file, data identifying said requesting computer and a storage location of said computer file.

75. Apparatus as claimed in any one of claims 72 to 74, wherein if said file access clearance response indicates a scan of said computer file is required by said assessment computer, then a computer file transmitter is operable to transmit said computer file from said requesting computer to said assessment computer, receiving at said assessment computer said computer file from said requesting computer and performing a malware scan of said computer file.

76. Apparatus as claimed in any one of claims 72 to 75, wherein if said file access clearance response indicates access to said computer file is denied, then triggering a denied access response in said assessment computer.

77. Apparatus as claimed in any one of claims 72 to 76, wherein if said file access clearance response indicates access to said computer file is denied, then triggering a denied access response in said requesting computer.

78. Apparatus as claimed in any one of claims 72 to 77, wherein said assessment computer stores a database of computer files previously assessed as to whether they contain malware.

79. Apparatus as claimed in claim 78, wherein said database includes for each computer file fields specifying one or more of a filename of said computer file, data identifying said requesting computer and a storage location of said computer file, a checksum value calculated from said computer file, an access flag indicating whether access to said computer file is denied and a persistence flag indicating whether entries relating to said computer file should be purged from said database during purge operations.

80. Apparatus as claimed in any one of claims 72 to 79, wherein said assessment computer is operable in at least a higher level security mode and a lower level security mode, said assessment computer serving to deny access to greater range of computer files when operating in said higher level security mode compared with said lower level security mode.

81. Apparatus as claimed in claim 80, wherein said assessment computer is triggered to change from said lower level security mode to said higher level security mode by a lock down trigger message received at said assessment computer from a remote computer.

82. Apparatus as claimed in any one of claims 72 to 81, wherein a plurality of requesting computers share access to an assessment computer for determining whether file access requests by those requesting computers should be denied

Trend Micro, Inc.
EX1007-548

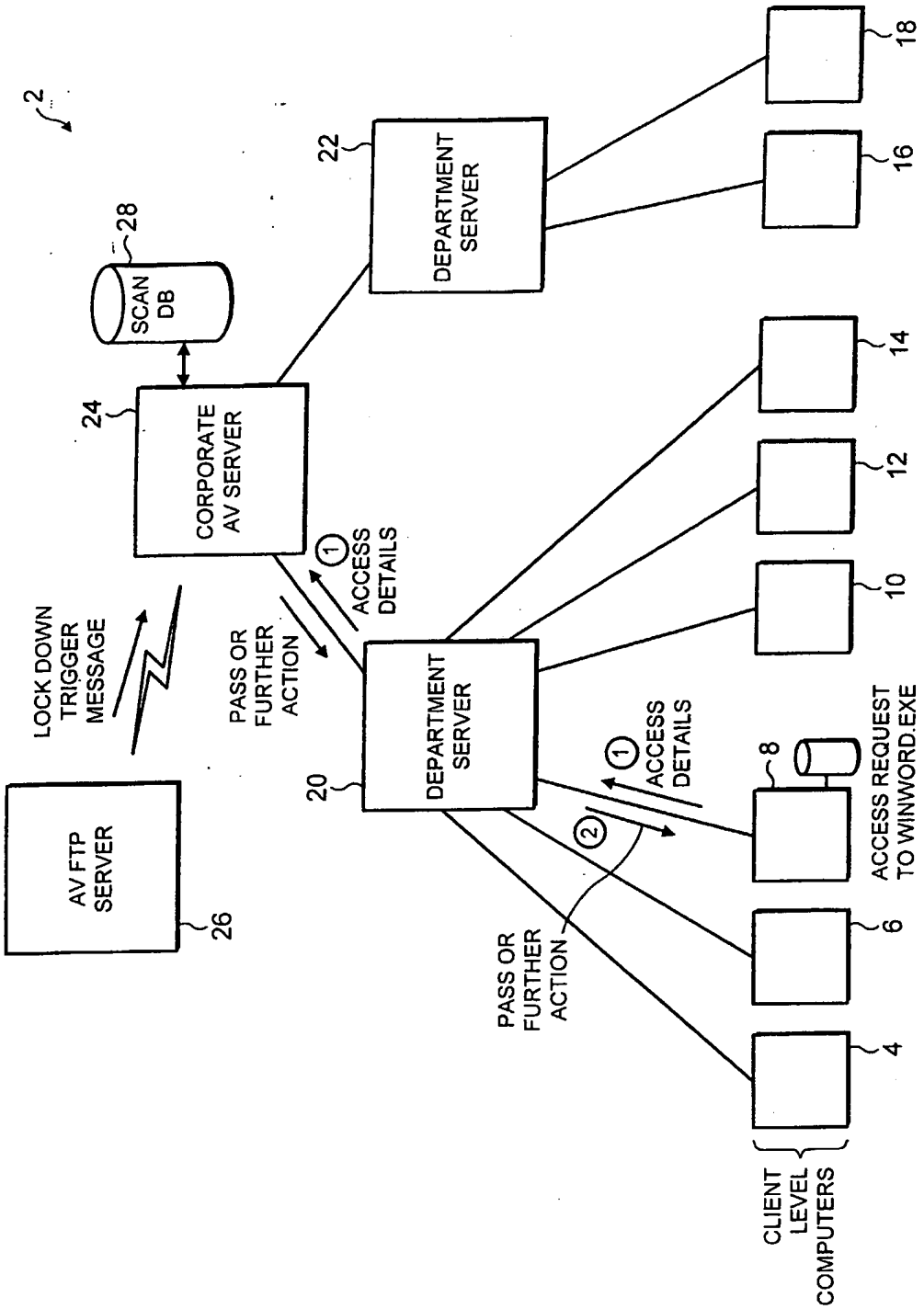


FIG. 1

Trend Micro, Inc.
EX1007-549

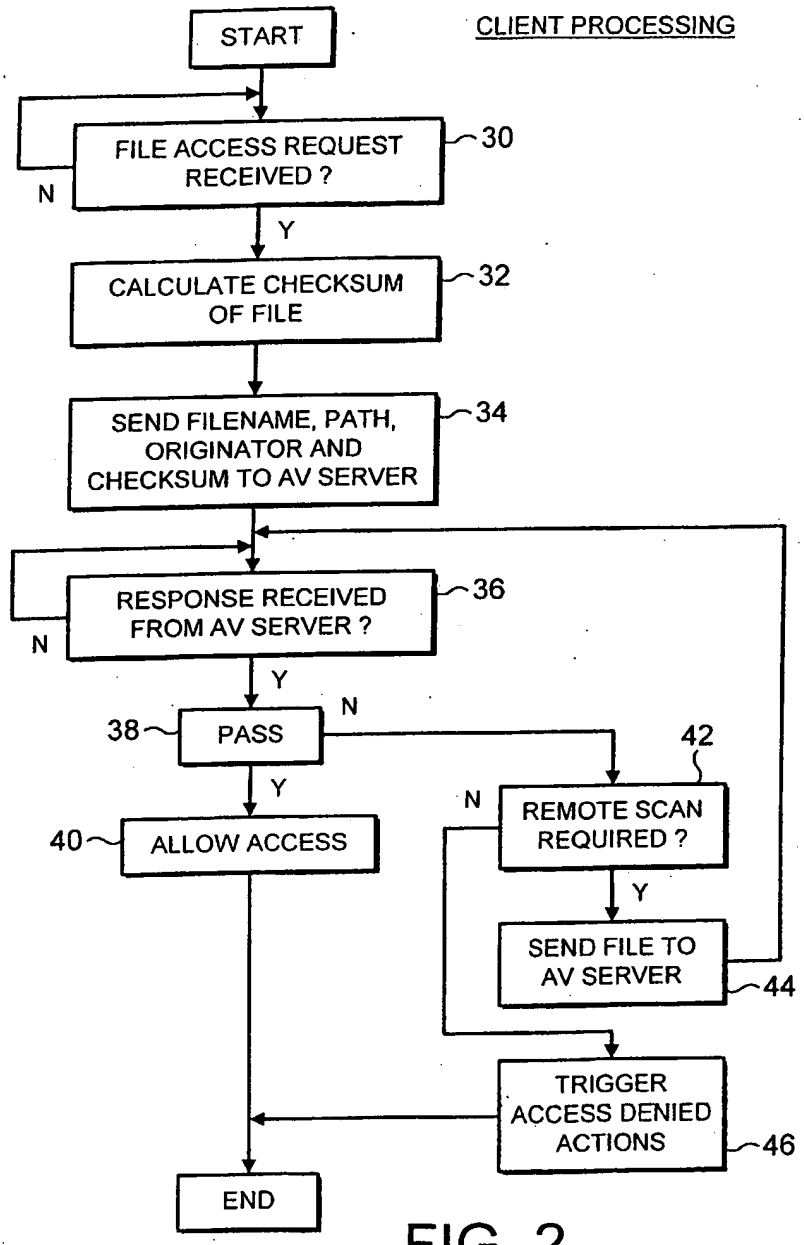


FIG. 2

Trend Micro, Inc.
EX1007-550

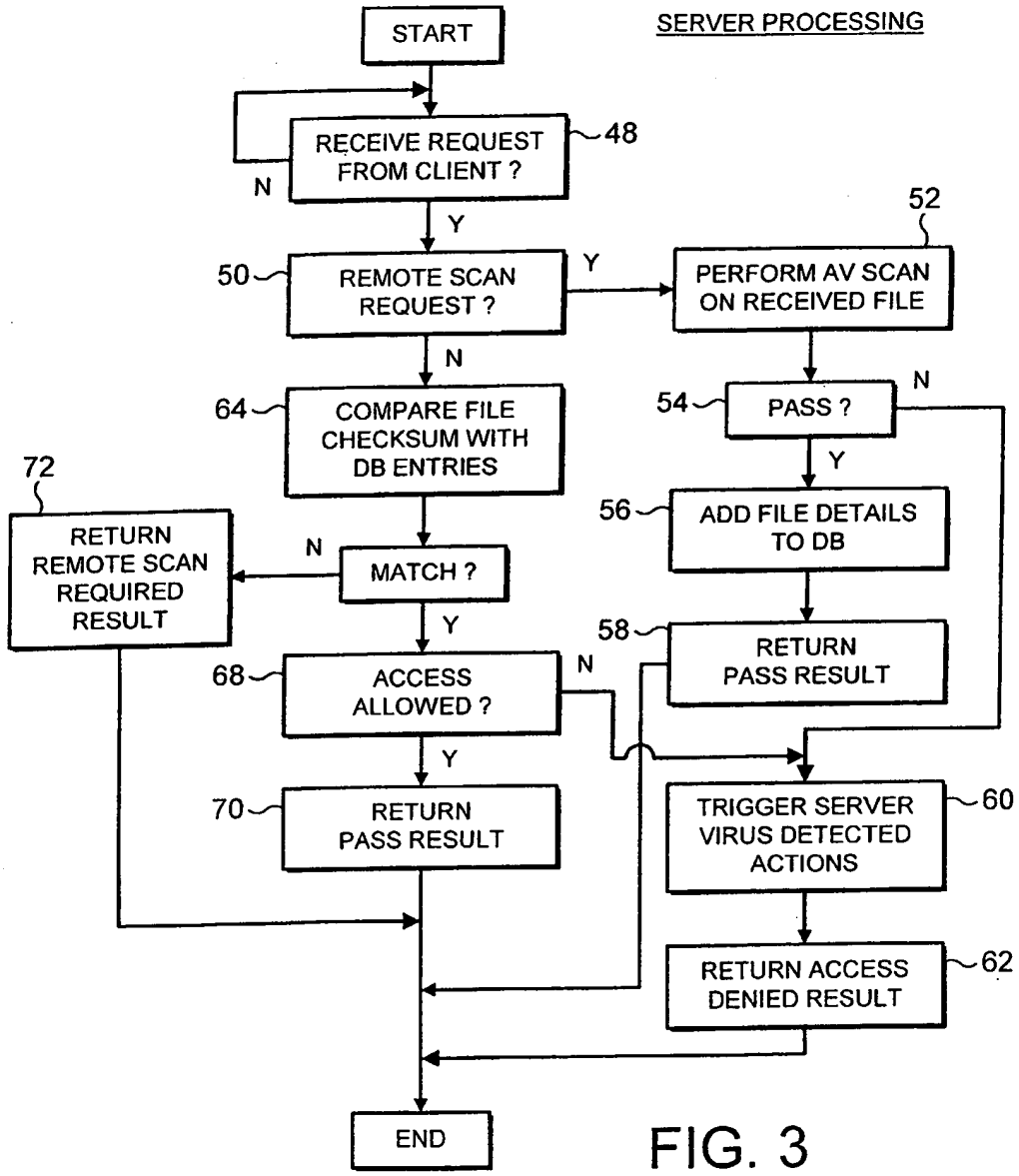


FIG. 3

Trend Micro, Inc.
EX1007-551

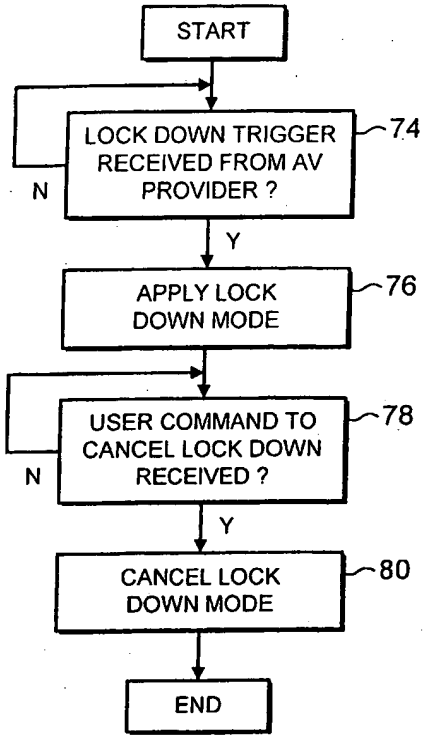


FIG. 4

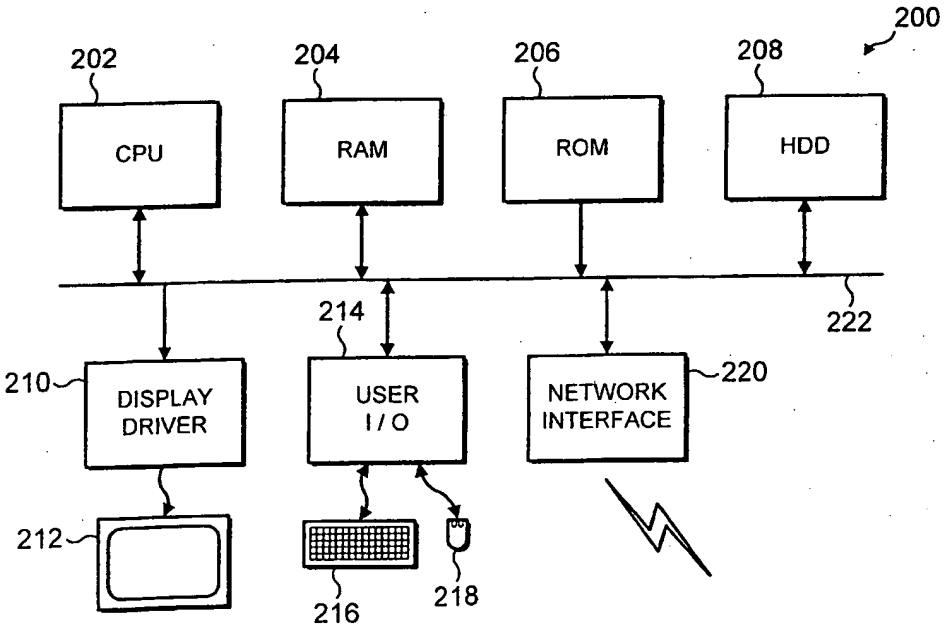


FIG. 6

Trend Micro, Inc.
EX1007-552

FILENAME	ORIGINATOR	PATH	CHECKSUM	ACCESS ALLOWED	PERSISTENT
WINWORD.EXE	CLIENT N	C:\PROGRAM FILES\ MICROSOFT OFFICE\ OFFICE	[12345678]	Y	Y

FIG. 5

Trend Micro, Inc.
EX1007-553

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 280 040 A3

(12) EUROPEAN PATENT APPLICATION

(88) Date of publication A3:
24.03.2004 Bulletin 2004/13

(43) Date of publication A2:
29.01.2003 Bulletin 2003/05

(21) Application number: **02253235.2**

(22) Date of filing: **09.05.2002**

(51) Int Cl.[7]: **G06F 1/00**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **26.07.2001 US 912392**

(71) Applicant: **Networks Associates Technology, Inc.**
Santa Clara, CA 95054 (US)

(72) Inventors:
- **Hinchliffe, Alexander James**
  Milton Keynes, Buckinghamshire MK7 7LL (GB)
- **Howard, Fraser Peter**
  Chipping Norton, Oxon OX7 3BT (GB)
- **Kemp, Andrew**
  Oxford, Oxon OX4 2QT (GB)
- **Rai, Bobby**
  Kettering, Northants NN15 5BP (GB)

(74) Representative:
Robinson, Nigel Alexander Julian
D. Young & Co.,
21 New Fetter Lane
London EC4A 1DA (GB)

(54) Centrally managed virus scanning system

(57) A malware detecting system is described in which a client computer 8 sends a file access clearance request to an assessment computer 24 to determine whether or not access is permitted to that file. In this way, the different client computers on a network 2 may share their scan results and centralised control may be provided over access permissions to individual files or groups of files.

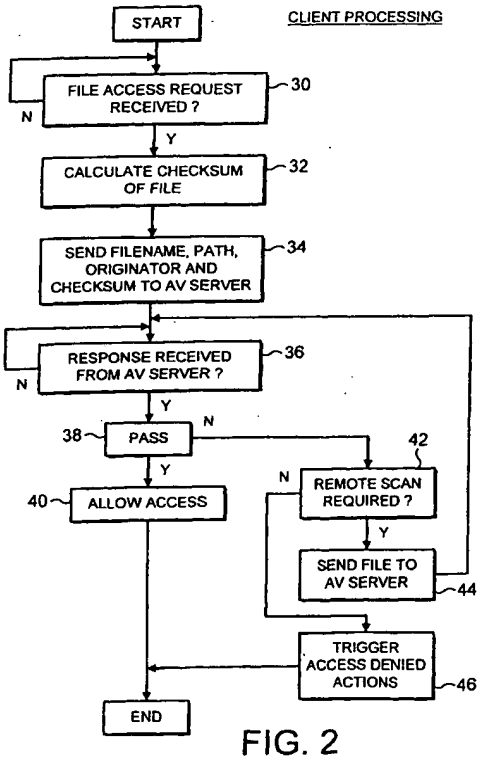


FIG. 2

EP 1 280 040 A3

Trend Micro, Inc.
EX1007-554

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 02 25 3235

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
X	US 6 195 587 B1 (HRUSKA JAN ET AL) 27 February 2001 (2001-02-27) * abstract * * column 1, line 1-9; figure 3 * * column 2, line 15-65; figures 1,2 * * column 3, line 5-25,40-67 * * column 5, line 1-60 * * column 4 * ---	1-13, 16-24, 28-41, 44-52, 55-68, 71-79,82	G06F1/00 G06F21/00
X	US 5 960 170 A (LAU STEVEN YUEN-LAM ET AL) 28 September 1999 (1999-09-28) * abstract * * column 1-10; figures 1-8 * * column 11, line 1-55 * * column 12-27 * ---	1,7,17, 29,35, 45,56, 62,72	
X	EP 1 111 507 A (SECURE OYJ F) 27 June 2001 (2001-06-27) * the whole document * ---	1,7,17, 29,35, 45,56, 62,72	
	-/--		

TECHNICAL FIELDS SEARCHED (Int.Cl.7)
G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
MUNICH	26 January 2004	Nazzaro, A

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

Trend Micro, Inc.
EX1007-555

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 02 25 3235

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
A	MARMELSTEIN R E ET AL: "A distributed architecture for an adaptive computer virus immune system" SYSTEMS, MAN, AND CYBERNETICS, 1998. 1998 IEEE INTERNATIONAL CONFERENCE ON SAN DIEGO, CA, USA 11-14 OCT. 1998, NEW YORK, NY, USA,IEEE, US, 11 October 1998 (1998-10-11), pages 3838-3843, XP010310923 ISBN: 0-7803-4778-1 * abstract; figure 1 * * page 3841 * * page 3842 * -----	14,15, 25,26, 42,43, 53,54, 69,70, 80,81	

TECHNICAL FIELDS SEARCHED (Int.Cl.7)

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
MUNICH	26 January 2004	Nazzaro, A

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

Trend Micro, Inc.
EX1007-556

European Patent Office

Application Number
EP 02 25 3235

CLAIMS INCURRING FEES

The present European patent application comprised at the time of filing more than ten claims.

☐ Only part of the claims have been paid within the prescribed time limit. The present European search report has been drawn up for the first ten claims and for those claims for which claims fees have been paid, namely claim(s):

☐ No claims fees have been paid within the prescribed time limit. The present European search report has been drawn up for the first ten claims.

LACK OF UNITY OF INVENTION

The Search Division considers that the present European patent application does not comply with the requirements of unity of invention and relates to several inventions or groups of inventions, namely:

see sheet B

☒ All further search fees have been paid within the fixed time limit. The present European search report has been drawn up for all claims.

☐ As all searchable claims could be searched without effort justifying an additional fee, the Search Division did not invite payment of any additional fee.

☐ Only part of the further search fees have been paid within the fixed time limit. The present European search report has been drawn up for those parts of the European patent application which relate to the inventions in respect of which search fees have been paid, namely claims:

☐ None of the further search fees have been paid within the fixed time limit. The present European search report has been drawn up for those parts of the European patent application which relate to the invention first mentioned in the claims, namely claims:

Trend Micro, Inc.
EX1007-557

European Patent Office

LACK OF UNITY OF INVENTION
SHEET B

Application Number
EP 02 25 3235

The Search Division considers that the present European patent application does not comply with the requirements of unity of invention and relates to several inventions or groups of inventions, namely:

1. Claims: 1-13,16-24,28-41,44-52,55-68,71-79,82

Computer program , method and apparatus for detecting malware, comprising: detecting logic, file access clearance request generating and transmitting logic, file access clearance receiving logic at an assessment computer, file access clearance response generating and transmitting logic, file access clearance response receiving logic and file access permitting logic if file access clearance response indicates said file does not contain malware.

2. Claims: 14-15, 25-26, 42-43, 53-54, 69-70, 80-81

Computer program method and apparatus for detecting malware comprising detecting logic, file access clearance request generating and transmitting logic, file access clearance receiving logic at an assessment computer, file access clearance response generating and transmitting logic, file access clearance response receiving logic and file access permitting logic if file access clearance response indicates said file does not contain malware. The assessment computer is operable in a higher level security mode serving to deny access to greater range of computer files. The change from lower to higher level security mode is triggered from a remote computer.

Trend Micro, Inc.
EX1007-558

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 02 25 3235

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

26-01-2004

Patent document cited in search report		Publication date		Patent family member(s)	Publication date
US 6195587	B1	27-02-2001	GB	2283341 A	03-05-1995
			AU	8000394 A	22-05-1995
			DE	69407812 D1	12-02-1998
			DE	69407812 T2	16-07-1998
			EP	0725951 A1	14-08-1996
			WO	9512162 A1	04-05-1995
			JP	9504395 T	28-04-1997
US 5960170	A	28-09-1999	AU	6537698 A	12-10-1998
			EP	0968466 A1	05-01-2000
			JP	2001515625 T	18-09-2001
			WO	9841919 A1	24-09-1998
EP 1111507	A	27-06-2001	GB	2353372 A	21-02-2001
			EP	1111507 A2	27-06-2001
			US	2001005889 A1	28-06-2001

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

Trend Micro, Inc.
EX1007-559

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 549 012 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
29.06.2005 Bulletin 2005/26

(51) Int Cl.[7]: **H04L 29/06**, G06F 1/00

(21) Application number: **03447310.8**

(22) Date of filing: **24.12.2003**

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LI LU MC NL PT RO SE SI SK TR
Designated Extension States:
AL LT LV MK

(71) Applicant: **DataCenterTechnologies N.V.**
9080 Lochristi (BE)

(72) Inventor: **De Spiegeleer, Kristof**
8301 Knokke-Heist (BE)

(74) Representative: **Bird, William Edward et al**
Bird Goen & Co.,
Klein Dalenstraat 42A
3020 Winksele (BE)

(54) Method and system for identifying the content of files in a network

(57) A method and system for performing securing and controlling of a network using content identification of files in a network having a central infrastructure and local computing devices is presented. The method comprises calculating a hash value of a new file created or received on a local computing device, transmitting the hash value to the central infrastructure, comparing the hash value with a previously determined hash value stored in a database on the central infrastructure to determine whether the file is new to the network and if the file is new to the network, checking the file content with a content identifying engine, installed and updated on the central infrastructure. Content attributes are determined for the files which allow to perform appropriate actions on the local computing devices according to policy rules.

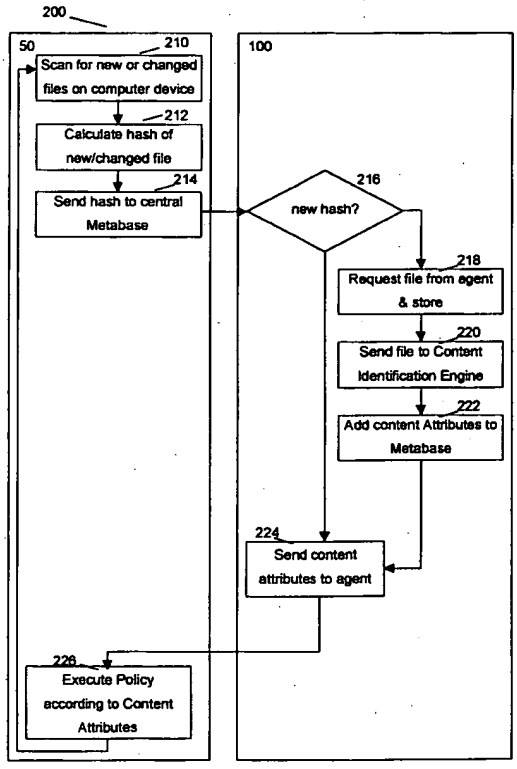


Fig. 3

EP 1 549 012 A1

Trend Micro, Inc.
EX1007-560

Description

Technical field of the invention

[0001] The invention relates to a method and system to control the content of computer files, e.g. containing text or graphical data. More specifically, a method and system is described for checking and managing the security status and the content of computer files on a local computing device in a network environment.

Background of the invention

[0002] In today's world, computers are widely spread. Very often, especially in business environment, they are interconnected in small or larger networks. As software and data often are an important part of the investment goods of both private persons and firms, it is important to protect single computing devices and complete networks and their workstations against attacks from viruses, trojan horses, worms and malicious software. Another problem is related to the amount of files containing undesirable content such as explicit adult content. These files are often received on local computing devices uninvited and unwanted.

[0003] To solve the security problems associated with viruses, virus protection systems, also called virus checkers, have been developed. Some examples of conventional virus checkers are Norton AntiVirus, McAfee VirusScan, PC-cillin, Kaspersky Anti-Virus. Most of these conventional virus protection software packages can be configured so that they are continuously running in the background of the computing device and providing continuous protection. These virus protection systems compare codes of new or amended software with fingerprints (e.g. parts of code introduced in files by the viruses) of well known viruses. Other virus protection systems compare codes of all data available on the computing device. This leads to the use of a significant amount of central processing unit (CPU) time, which limits the capacity of the computing device for performing other tasks. Furthermore, the working principle of these virus checkers makes these software packages work rather reactive than proactive, as the fingerprint of the virus needs to be known in order for a virus scanning program to recognise it. This implies that the database of fingerprints needs to be updated very regularly in order to be secured against relatively new viruses. Consequently, the secure state of the computer is not only depending on external factors like the accurateness with which fingerprints of new viruses are made available by the suppliers of virus protection software packages, but also on the sense of duty of the user regarding performing updates regularly. If updates are provided centrally from a server automatically, then network capacity is reduced as these virus updates must be sent to each workstation.

[0004] In a network environment, the problem of updating such a database of fingerprints becomes significantly more important, as it implies that the responsibility is put to all users, who all have to update their virus checker database. Alternatively, the virus scanning could be performed by a central server, thus limiting the updating for new fingerprints to the central server. Nevertheless this implies that a large amount of data needs to be transferred over the network on a regular basis thereby utilising large amounts of expensive network bandwidth and possibly (depending on the number of clients for the server) overloading the network or server capacity for other activities.

[0005] In order to limit the amount of CPU time used, additional techniques have been developed to speed up the virus scanning process. These very often include hashing of the content of files. Hashing is one example of application of a "one-way-function". A one-way-function is an algorithm which when applied in one direction makes the reverse direction almost impossible to perform. A one-way-function generates a value such as a hash value by a calculation on the content of a file and can uniquely fingerprint this file if the one-way-function is complex enough to avoid duplicate values from different files. The uniqueness of a hashing function depends on the type of hashing function that is used, i.e. the size of the digest that is formed and the quality of function. Good hashing functions have the fewest collisions in a table, i.e. the chance of providing the same hash value for different files is the smallest. As mentioned, this is also determined by the size of the digest, i.e. hash value, that is calculated. If e.g. a 128-bit digest is used, the number of possible different values that can be obtained is 2^{128}.

[0006] It is known to use hashing for virus checking, possibly in a network environment. Typically, a hash of an application selected to run on a local computer is calculated, a stored hash from a database on a secured computer is retrieved on the local computer, whereby the secured computer can be a secured part of the local computer or a network server, and both values are compared. If there is a match, the application is executed, if there is no match, a security action is performed. This security action comprises loading a virus scanner on the local computer. It may also comprise alerting the network administrator. Furthermore it is also known to use this for differentiating accessibility to software from different workstations and as a way of checking whether software is licensed.

[0007] It is also known to use hashing in a method of identifying rogue software on a computer system or device. The method typically is applicable in a network environment. A hash value of a software application to be executed is calculated, this hash value is transferred to a server and compared with previously stored values. One of the essential features is that the method uses a database on a server, the server being a server with a large number of clients. The database on the server thereby is built up by adding information by different cli-

Trend Micro, Inc.
EX1007-561

ents so that most software applications and their corresponding fingerprints are already stored in the database. The database is built up by checking software applications on authenticity with the owners of the application. If this is not possible, the system is also able to give a heuristic result, evaluating the occurrence of this application on local computers from other clients.

[0008] Methods for sending an electronic file by electronic mail, i.e. e-mail, including a file content and message content identifier are known. Depending on the message content identification, the message is delivered to a customer or not. The method may be used to organise e-mail delivery, but it has the disadvantage of being focussed on e-mail delivery and it does not allow to secure all files in a network.

[0009] Monitoring of electronic mail messages, to protect a computer system for protection against virus attacks and unsolicited commercial e-mail (UCE) is also known. Such a system is preferably installed on a mail server or an Internet Service Provider and checks specific parts of e-mails by calculating a digest and comparing it with stored digest values of e-mails previously received. In this way it is determined whether the e-mail has an approved digest or whether the e-mail is UCE or contains a e-mail worm. The system has the disadvantage that it is focussed on e-mail viruses and SPAM and that it does not allow to check all data files or executable files which are possibly infected, e.g. by files copied from external memory storage means like floppy disks or CD-ROMs or by e.g. Trojan horses.

[0010] Controlling the execution of software on different workstations according to certain policy rules by a network server is known, whereby an improved computer security system is obtained, by classifying software. It is suggested that this classification can be based on several forms of data one of which is e.g. the hash values of software data. This typically is performed by the calculation of hash values of a program if it is selected for loading and execution, and comparison of the hash value with a trusted value to determine the rule of execution. The classification also may be based on a hash of the content, a digital signature, the file system or network path or the URL zone.

[0011] The above mentioned methods and systems describe the use of hashing functions to check whether software applications are authentic or to regulate the execution of software applications. Nevertheless the problem of virus scanning all new files in a network using a conventional virus scanner whereby the necessity of updating the database of fingerprints of a conventional virus scanner on every local computer is limited is not discussed.

Summary of the invention

[0012] It is an object of the present invention to provide a system and a method for identifying the content of files new on local computing devices in a network while limiting the necessity of updating the information needed by a content identifier on every local computing device.

[0013] It is an advantage of the present invention that, if the invention is used as a virus checker, the security level is further increased as the database of fingerprints of a conventional virus scanner does not have to be updated on every local computing device.

[0014] It is a specific advantage of the present invention that the content of a file new to a network is only identified once for the whole network.

[0015] It is furthermore a specific advantage of the present invention that the total processor (CPU) processing time in the network and the amount of network traffic is reduced.

[0016] The above described object and advantages are obtained with a method and system of content identification in a network according to the present invention.

[0017] The method for identifying the content of a data file in a network environment is used for a network having at least one local computing device linked to a remaining part of the network environment including a central infrastructure. The method and system comprises calculating a reference value for a new file on one of said at least one local computing devices using a one-way-function, transmitting said calculated reference value to said central infrastructure, comparing said calculated reference value with reference values previously stored within the remaining part of the network environment.

[0018] The method further comprises, after comparing, deciding that the content of the new file is already identified if a match between said calculated reference value and a previously stored reference value is found and retrieving the corresponding content attributes; or deciding that the content of the new file is not yet identified if no match between said calculated reference value and any of the previously stored reference values is found, followed by said central infrastructure identifying the content of said new file and determining content attributes corresponding with the content of the new file and storing a copy of said content attributes, after deciding, triggering an action on said local computing device in accordance with said content attributes.

[0019] In the method for identifying the content of a data file in a network environment, the reference value may be a hash value. The reference values previously stored may be stored within the central infrastructure. In the method and system for identifying the content of a data file in a network environment, identifying the content of the new file may comprise scanning the new file for viruses using an anti-virus checker means on a central infrastructure.

[0020] The method may furthermore comprise transferring the new file from the local computing device to the central infrastructure before said identifying the content of said new file is performed. Furthermore it may comprise storing a copy of the new file on the central

Trend Micro, Inc.
EX1007-562

infrastructure after transferring the new file from the local computing device to the central infrastructure. An address of where the file is stored may be stored together with the hash value, as to be able to quickly track copies of the files stored on the central infrastructure.

[0021] In the method of the present invention, triggering an action on the local computing device in accordance with said content attributes may comprise replacement of the new file on the local computing device with a copy of a previous version of said new file. Furthermore, triggering an action on the local computing device in accordance with said content attributes may also comprise replacement of the new file on the local computing device with another version of said new file restored from the remaining part of the network environment.

[0022] The method of the present invention furthermore may comprise sharing the new file on the local computing device to the central infrastructure before said identifying the content of said new file is performed and whereby said identifying the content of said new file is performed by remotely identifying the content over the network environment. The method may comprise checking the functioning of the local agent on the local computing device.

[0023] Furthermore, triggering an action on the local computing device may be performed after transmitting the content attributes corresponding to the new file to the local computing device.

[0024] In the method for identifying the content of a data file in a network environment, identifying the content of the new file may comprise one or more of the group of scanning for adult content, scanning for Self Promotional Advertising Messages or Unsolicited Commercial E-mail (UCE) and scanning for copyrighted information. The method may further relate to a method and system for providing a content firewall, whereby one local computing device is connected to the external network, which may e.g. be the internet, and the one local computing device is also connected to the network environment formed by the remaining local computing devices. The one local computing device thus links the network environment with an external network and is the only computing device that is directly connected to sources external from the network environment. The local computing device thus acts as a content firewall as to protect the network environment from attacks originating from places in the external network. The local computing device may act as a content firewall working in a promiscuous way, i.e. whereby the local computing device acts as a content firewall that sees all traffic passing by, executes the hashing and comparing functions and contacts the agents to enforce a policy.

[0025] The method may be specifically related to a method for checking the security status of a network and its components. In this embodiment, a method for determining the security status of a data file in a network environment is used in a network having at least one local computing device linked to a remaining part of the network environment including a central infrastructure. The method comprises calculating a reference value for a new file on one of the at least one local computing devices using a one-way-function, transmitting said calculated reference value to said central infrastructure, comparing said calculated reference value with reference values previously stored within the remaining part of the network environment and after comparing, deciding that the security status of the file has already been checked if a match between the calculated reference value and a previously stored reference value is found and retrieving the corresponding security status; or deciding that the security status of the new file is not yet identified if no match between said calculated reference value and any of the previously stored reference values is found, followed by said central infrastructure checking the security status of the new file and determining the security status corresponding with the new file and storing a copy of the security status, followed by
after deciding, triggering an action on said local computing device in accordance with the security status of the new file. This action may be e.g. making the file inaccessible for the user of the local computing device and for other users in the network or restoring the infected file.

[0026] The methods described above may be triggered by an action performed on the local agent. The triggering by an action performed on the local agent may be e.g. running an application or opening a file.

[0027] The invention also relates to a method for altering a system for identifying the content of a file in a network environment according to the systems described above, the network environment comprising means for calculating a one-way function, at least one local computing device linked to a remaining part of the network environment including a central infrastructure and means for identifying the content and, the method comprising altering said means for identifying the content or said means for calculating a one-way function, scanning the remaining part of the network environment for reference values calculated with a one-way function and for each of the reference values, requesting a file that corresponds with said reference value from said network environment, sending the file to means for identifying the content, identifying the content of said file and determining content attributes corresponding with the content of the file and storing a copy of said content attributes, sending the content attributes to every local computing device containing the file and after sending; triggering an action on said local computing device in accordance with said content attributes. Altering of a system for identifying the content of a file in a network environment may be triggered by the introduction of a new one-way function to calculate reference values or may be also triggered by the updating of the means for identifying the content of the files. The method may be used to scan only part of the hashing keys in the remain-

Trend Micro, Inc.
EX1007-563

ing part of the network environment, e.g. hashing keys of files of which the content is identified after a certain date, as to minimise the actions to be performed. The date of the previous content identification may be retrieved from the content attributes.

[0028] The reference values may be hashing values. The means for identifying the content may be an anti-virus checker means, a means for scanning for adult content, a means for scanning for Self Promotional Advertising Messages or a means for scanning for copyrighted information. Triggering an action on the local computing device in accordance with said content attributes may comprise replacement of the file on the local computing device with another version of the file restored from the remaining part of the network environment or may comprise replacement of a file with a copy of a previous version of the file or may comprise putting the file in quarantine or removing the file.

[0029] The invention is also related to a computer program product for executing any of the above described methods, when executed on a network. The invention furthermore relates to a system for identifying the content of a file in a network environment, said network environment comprising at least one local computing device linked to a remaining part the network environment which includes a central infrastructure and, said remaining part including a stored database, whereby the system comprises means for calculating a reference value for a new file on said local computing device using a one-way-function, means for transmitting said calculated reference value to said central infrastructure and means for comparing said calculated reference value with previously stored reference values from the database. The system furthermore comprises means for deciding whether the content of the new file is already identified based on comparison of said calculated reference value and reference values previously stored within the remaining part, means located on the central infrastructure, for identifying the content of the new file and as to assign content attributes if the new file has not been identified yet and means for storing said content attributes within the remaining part, and means for triggering an action on said local computing device in accordance with content attributes for said new file.

[0030] In the system according to the present invention, the means for identifying the content of a file may comprise an anti-virus checker means on said central infrastructure. Furthermore, means for storing a copy of the new file within the remaining part. The means for identifying the content of a file may comprise one or more of the group of means for scanning for adult content, scanning for Self Promotional Advertising Messages and scanning for copyrighted information.

[0031] The invention may also relate to a machine readable data storage device, storing the computer program product for executing any of the above described methods, when executed on a network. Furthermore, the invention may also relate to the transmission of the computer program product for executing any of the above described methods.

[0032] Although there has been constant improvement, change and evolution of methods of virus scanning and content identification of data files, the present concepts are believed to represent substantial new and novel improvements, including departures from prior practices, resulting in the provision of more efficient, stable and reliable methods of this nature.

[0033] These and other characteristics, features and advantages of the present invention will become apparent from the following detailed description, taken in conjunction with the accompanying drawings, which illustrate, by way of example, the principles of the invention. This description is given for the sake of example only, without limiting the scope of the invention. The reference figures quoted below refer to the attached drawings.

Brief description of the drawings

[0034]

Fig. 1 is a schematic representation of a computer network

Fig. 2 is a schematic representation of a central infrastructure and its basic software components

Fig. 3 is a schematic representation of a local agent-driven content identification process.

Fig. 4 is a schematic representation of a metabase-driven content identification process.

Fig. 5 is a schematic representation of a computer network to which the content firewall system and method can be applied.

[0035] In the different figures, the same reference figures refer to the same or analogous elements.

Description of illustrative embodiments

[0036] The present invention will be described with respect to particular embodiments and with reference to certain drawings but the invention is not limited thereto but only by the claims. The drawings described are only schematic and are non-limiting. In the drawings, the size of some of the elements may be exaggerated and not drawn on scale for illustrative purposes. Where the term "comprising" is used in the present description and claims, it does not exclude other elements or steps.

[0037] Furthermore, the terms first, second, third and the like in the description and in the claims, are used for distinguishing between similar elements and not necessarily for describing a sequential or chronological order. It is to be understood that the terms so used are interchangeable under appropriate circumstances and that the embodiments of the invention described herein are capable of operation in other sequences than described or illustrated herein.

[0038] In this description, the terms "file", "program",

Trend Micro, Inc.
EX1007-564

"computer file", "computer program", "data file" and "data" are used interchangeably, and any one use may imply the other terms, according to the context used. The terms "hash" and "hashing" will be used as examples of the application of one-way-functions but the present invention is not limited to a particular form of one-way-function.

[0039] The term "computing device" should be interpreted widely to include any device capable of carrying out computations and/or executing algorithms. A computing device may be any of a laptop, workstation, personal computer, PDA, smart phone, router, network printer or any other device which has a processor and can be connected to a network such as e.g. faxing devices or copiers or any dedicated electronic device such as a so-called "hardware firewall" or a modem.

[0040] The method and system to secure and control a network by identifying the content of each new file in the network can be used on any type of network. This may be a private network which may be a virtual private network, a local area network (LAN) or a wide area network (WAN). This may also be within a part of a public wide area network such as the internet. If a part of a public wide area network is used, this may be performed by remotely providing the method and system for identifying the content of each file by a service provider using an ASP or XSP business model, wherein the central infrastructure is provided to a paying client operating a local computing device. An exemplary network 10 is shown in figure 1, showing several local computing devices 50a, 50b, ... , 50i and a central infrastructure 100, also called a server. The number of local computing devices 50 connected to the network 10 is not limiting for the method of securing and controlling a network 10 according to the current invention. In business environment this number of local computing devices 50 typically ranges from a few to a few thousands. The method and system for identifying the content of each new file present in the network 10 may be used with many different operating systems such as Microsoft DOS, Apple Macintosh OS, OS/2, Unix, DataCenter-Technologies' Operating Systems, ...

[0041] In order to provide a quick method of securing and determining content identification of files, the method and system according to the present invention will determine hash values of new files present on the local computing devices 50, compare them with previously stored hash values and file information on a central server and determine the content of files new to the network 10 using a content identifying engine on the central infrastructure 100. The content attributes describing the content of a new file are then send to the local computing device 50 where an appropriate action is performed. It is also possible that the content attributes are not sent to the local computing device 50 but that the appropriate action is triggered from the central infrastructure 100. New files typically are files wherein new content has been generated on a local computing device 50 or when an external file has been received. The wording "file" may refer to data as well as to software applications, also called software.

[0042] The central infrastructure 100 contains a database, also called metabase 110, which contains a record for every hash value that is calculated for a file that already exists on one of the local computing devices 50. Besides the hash value, this record also contains a number of other fields. In these fields, file source information is stored. The file source information corresponding with a specific hash value includes the file name, a list of local computing devices 50 where the files that correspond to this hash value are residing on, including the path to the file on the file system of the local computing devices 50 and the date of last modification. An example of file source information for a specific file is given in Table 1.

Table 1

Filename	Myexampleword.doc
Path	c:\data\
Assetname	PCmarketing001
ModDate	23/4/2002

[0043] In a further field, a list of content attributes that identifies the type of content that is enclosed by the file is stored. The content attributes can e.g. refer to a file that contains a virus, a file that is a copyrighted MP3 audio file, a file that is a copyrighted video file, a file that is a picture, a file that is a picture that might contain adult content, a file that is a Self Promotional Advertising Message (SPAM), a file that is a HOAX, a file containing explicit lyrics or a file containing pieces of executable code. This list is not limiting.

[0044] The central infrastructure 100 furthermore contains a content identification engine 120. This can be a software application 130 or a set of software applications 130a, 130b, 130c, 130d, ... that use the content of a file to determine which type of content the file contains. These software applications may be various :

- a virus scanner : this is a piece of software that scans the content of the presented file and compares it with a database of known fingerprints of viruses. This can be any conventional virus scanning software like e.g. Norton anti-virus by Symantec Corporation, McAfee by Network Associates Technologies Inc., PC-cillin by Trend Micro, Kapersky Anti-Virus by Kaspersky Lab, F-secure Anti-Virus by F-Secure Corporation, ...
- an adult content in pictures scanner : this is a piece of software that scans the content of the presented file for the presence of shading, colors, textures that might represent adult content. Scanning pictures for adult content is already known. Content adult can e.g. be determined by the amount of nude that is

Trend Micro, Inc.
EX1007-565

shown. Skin tones have hue-saturation values that are in a specific range. Therefore, if an image is scanned, it is possible to determine the amount of pixels having a skin tone character and to compare it with the total number of pixels. The ratio of skin tone pixels to the total number of pixels allows to determine a ratio of possible adult content in an image. Thresholds often are introduced so that images can be classified according to their possible adult content. In similar way, video images can be categorised, whereby the video is split into its different frames and wherein the images are categorised according to the above method.

- A scanner for internet content ratings : A piece of software that scans objects for adult content based on the PICS, i.e. the Platform for Internet Content Selection, label system. On voluntary basis, internet content providers can provide internet objects with a PICS rating determining the adult content in the internet object. This PICS rating is stored in the meta data of the object. This data normally is not visible to the viewer of an internet object. The rating systems is well known and an example of a scanner for internet content ratings is provided in the Netscape web browser for scanning the content of web pages.
- A scanner for scanning an object for explicit lyrics which may indicate adult content. This is known for both text files and audio files. Audio files are first transferred to text files. Subsequently, the text files are scanned and compared with databases which contain explicit lyrics.
- a SPAM-engine: A piece of software that scans the content of e-mail messages for the presence of unalleged SPAM. Algorithms to recognize SPAM area already known. These are typically based on decomposing the text in an electronic mail message, associating statistics with the text using a statistical analyzer and coupling a neural network engine to the statistical analyzer to recognize unwanted messages based on statistical indicators.

Other examples of software applications that could be used in the content identification engine 120 are e.g. engines that scan for copyrighted content and that compares the content of the file to a database of copyrighted information, etc. In some adoptions, a human operator can pursue the role of content identification engine 120, where he manually tags a file with a content identification attribute. When the content identification engine 120 is activated, it will take a file from the local agent as input and produce a set of attributes that represent the detected content.

[0045] The content Identification engine 120 will thus be constructed as a piece of software aggregating the functionality of a set of third party engines.

[0046] In a further embodiment of the invention, a system and method in accordance with the above embodiment is described whereby the record corresponding with a specific hash value stored in the metabase 110 also comprises a field wherein the location of the file on the central infrastructure 100 corresponding with the hash value is stored. In this embodiment, a copy of all different files present on the local computing devices 50 in the network 10 is stored on the central infrastructure 100. So, the central infrastructure 100 of this embodiment also comprises a large amount of storing space. This preferably is a secured part of the central infrastructure 100, not directly connected to the network 10 so that these identical copies of the files present on the local computing devices 50 can be used in case the files on the local computing devices 50 are corrupted e.g. by a virus.

[0047] The hash value of the files are calculated using a hashing function. A hashing function typically is a one way function, i.e. given the digest, it is at least computationaly prohibitive to reconstruct the original data. Different types of hashing functions could be used : MD5, SHA-1 or ripemd all available from RSA Data Security Inc., haval which is designed at the University of Wollongong, snefru which is a Xerox secure hash function, etc.. The hashing functions most often used are MD5 and SHA-1. The MD5 algorithm takes as input a message of arbitrary length and produces as output a 128-bit "fingerprint" or "message digest" of the input. It is conjectured that it is computationally infeasible to produce two messages having the same message digest, or to produce any message having a given pre-specified target message digest. The MD5 algorithm is intended for digital signature applications, where a large file must be 'compressed' in a secure manner before being encrypted with a private (secret) key under a public-key cryptosystem. The MD5 algorithm is designed to be quite fast on 32-bit machines. In addition, the MD5 algorithm does not require any large substitution tables; the algorithm can be coded quite compactly. An alternative hashing function SHA-1, i.e. Secure Hashing Algorithm - 1, is a hashing algorithm generating a 160-bit hash. Newer versions of this algorithm also provide bit lengths of 256 and 512.

[0048] In the above mentioned embodiments describing a method and system to secure and/or control the network 10, a local agent is installed on the local computing device 50. The local agent is a piece of software that is running on a local computing device 50 and that performs certain algorithms and procedures. The local agent on the local computing device 50 is triggered typically in situations where new content is being generated on local computing devices 50. In order to avoid unnecessary hash value calculations and data transfer, a policy is setup to determine which actions will trigger the local agent and which actions do not trigger it. If e.g. a text document is being created, it is not necessary to check the file every time the document is saved. The policy for such a type of documents would preferably be that the document is checked e.g. if the file is both saved

Trend Micro, Inc.
EX1007-566

and closed. Some examples of actions which could trigger the local agent and thus start the content identification process are opening or receiving e-mail messages, opening or receiving e-mail attachments, running executable files, running files with dll or piff extension ... Applying this policy thus allows to prevent from continuously checking and scanning of documents leading to a limitation of the number of unnecessary hash calculations and content identification operations and thus limiting the unnecessary use of CPU time and load on the network traffic. The method and system of content identification is not limited due to the type of application in which the file is made.

[0049] The content identification process can be either triggered by the local agent on the local computing device 50 or it can be triggered by the central infrastructure 100. The latter process typically occurs in situations wherein new algorithms or tools are being used for content identification. Such new algorithms or tools can either be optimised algorithms and tools or previously uninstalled tools. Some examples of these tools, without restricting to these functions, could be virus checking, checking whether a file is a copyrighted MP3 Audio File, checking whether a file is a copyrighted Video File, checking whether a file is a picture that might contain adult content, checking whether a file is tagged as being SPAM or HOAX, checking whether a file contains explicit lyrics or checking whether a file contains copyrighted pieces of executable code. Updating of these tools may influence the status of the files and thus may in principle influence the corresponding records in the metabase 110. Therefore, depending on the type of update of the content identification means 120, it may be interesting to update the corresponding records.

[0050] In a specific embodiment, the method relates to a virus checker for a network environment. The networks 10 on which this method can be applied are the same as those described for the previous embodiments. The local agent calculates the hash value of a new file on the local computing device 50. This new file may comprise new content generated on the local computing device 50 or an external file which is received on the local computing device 50. The hash value of the new file and the corresponding file information then is sent to a central infrastructure 100, also called server, where it is compared to previously stored hash values corresponding with files that are already present on the different local computing devices 50 of the network 10. This comparison allows to check whether the file is new or not in the entire network 10. Alternatively, the hash value may also be first compared with a local database of hash values and file information corresponding with the files present on that particular local computing device 50 and subsequently, if the file has been found not yet present on the local computing device 50, the hash value and the corresponding file information may be interchanged with the central infrastructure 100 so it can be checked whether the file is new or not in the entire network 10. Although transferring the file information and the hash value of every new file only corresponds with a very small fraction of the network traffic for a conventional central virus checker, this alternative could reduce the network traffic used for virus checking even further. If a hash value has been identified as new on the network 10, the metabase agent triggers the local agent to transfer the file corresponding with the new hash value from the local computing device 50 to the central infrastructure 100. The transferring of the file may be performed in a secured way, i.e. the file may be transferred such that it cannot be influenced by a virus present at a network connection or such that, it if contains a virus, this cannot be spread over the whole network 10. To obtain this, a known secure transmission route, a tunnel and/or known session ecryption/decryption techniques may be used. A conventional virus checker, installed and updated on the central infrastructure 100 then checks the file for viruses. This can be any conventional virus checker like e.g. Norton anti-virus by Symantec Corporation, McAfee by Network Associates Technologies Inc., PC-cillin by Trend Micro, Kapersky Anti-Virus by Kaspersky Lab, F-secure Anti-Virus by F-Secure Corporation, ...

[0051] A specific advantage of the above described embodiments in the current invention is that the virus scanning software does not need to be updated on every local agent but that this is restricted to updating of the virus scanning software of the central infrastructure 100. In this way the security level of the network 10 is increased significantly as the security does not depend on the punctuality of the different users of the network 10 to update their virus scanning software. If the scanned file has no virus it will be marked in the metabase 110 as being a virus free file. If there is a virus found in a file the file will be marked as dangerous. A query will happen to the metabase 110 to find all files over the network 10 having the same corrupted hashing key. The result is a list of files with path, and assetname where the file is located. This information can be used to do actions to eleminate the danger of found viruses on all local computing devices 50, i.e. all workstations, from the complete network 10. In this way proactive virus scanning can be performed on other local computing devices 50, based on a virus detection on a first local computing device 50. Depending the policy defined for virus checking, the virus engine will inform an agent installed on the affected system to remove the file and if possible replace with either a recovered version delivered by the virus engine located on the central infrastructure 100, or a previous version of the file which didn't have the virus yet. The latter can be done easily by searching the metabase for a previous version of that file, or it can be performed by searching an uninfected version on another local computing device 50. If an uninfected version cannot be retrieved from either another local computing device 50 or the metabase residing on 100, the virus scanner should have a feature which allows it to save a

Trend Micro, Inc.
EX1007-567

new disinfected copy of the file on the central infrastructure 100. These advantages are also present for other content identifying packages.

[0052] In an alternative embodiment, if a file having a new hash value has been identified in the network 10, instead of transferring the file to the central infrastructure, the file may be automatically shared locally and a remote checker then may transfer a file-system which allows to check the file across the network 10 using the file sharing. The content tagging still is performed by the server. In order to improve security, the accessibility to the shared file is restricted to the server. Furthermore, a java applet could be transferred to the local agent to allow checking other files.

[0053] The previous embodiments are an improvement over a central virus checker which scans local computing devices 50 through the network 10. This is only possible if the local drives, e.g. C:\, D:\, ..., are shared. Besides the dangers of sharing with respect to security, the local user also easily can change the local sharing properties thereby preventing the remote checker from checking the files. This is at least partly avoided with the current invention as changing the network 10 sharing properties does not influence the operation of calculating the hash value of new files and sending it to the central infrastructure 100.

[0054] Another advantage is that it saves CPU time on the local computing device 50 as the CPU does not have to keep doing virus checking, it only has to calculate a one way function. It also saves network time : the administrating server does not have to update the virus checkers on the local computing devices 50 with virus updates, as a single central virus checker only is used and updated.

[0055] Fig. 3 shows a method 200 of the content identification process triggered by the local agent on the local computing device 50 according to the above mentioned embodiments. The different steps that occur during the process, both on a local computing device 50 and on the central infrastructure 100 are discussed.

[0056] The content identification process is based on continuously scanning for new data or applications on the local computing device 50 by the local agent. This scanning for data and applications is limited by the policy rule for determining when the local agent should be triggered, as described above. If a "new" file has been detected the method for securing and controlling the network 10 by content identification of new files is initiated. This is step 210. Method 200 then proceeds to step 212.

[0057] In step 212, a hash value of the "new" file is calculated using a hashing function like MD5 or SHA-1. This calculation is performed by using some CPU time of the local computing device. Nevertheless, the amount of CPU time used is drastically smaller than the CPU time that would be necessary if e.g. a conventional virus checker was used to check the file on the local computing device 50. Method 200 then proceeds to step 214.

[0058] In step 214, the hash value and the file source information is transferred from the local agent to the central infrastructure 100 of the network 10. If necessary, this transfer can be a secured transfer, whereby it is avoided that a virus which is positioned on a network connection changes both the file source information or the hashing key during transport of this data. Such a secured transmission can be made over a known secure transmission route, via a tunnel, or using known session ecryption/decryption techniques.

[0059] In step 216, the hash value is compared with the data already present in the metabase 110. As in the metabase 110, the hash values and file source information of all old files - i.e. every file that has been present on the network 10 and that is not "new" as described above - present in the network 10 are stored, it is possible to check whether the file already is present in the network 10. Therefore, if the hash value has been identified as new, this implies that the file is "new" for the whole network 10. If the file is new, method 200 proceeds to step 218. If the hash value is not new, this means that somewhere on a local computing device 50 in the network 10, the file does already exist. In this case, there already exists content attributes describing the content of the file. Method 200 then proceeds to step 224.

[0060] In step 218 the metabase agent triggers the local agent to transfer the file corresponding with the new hash value from the local computing device 50 to the central infrastructure 100. The transferring of the file may be performed in a secured way, i.e. the file may be transferred such that it cannot be influenced by a virus present at a network connection or such that, it if contains a virus, this cannot be spread over the whole network 10. To obtain this a known secure transmission route, a tunnel and/or known session ecryption/decryption techniques may be used. Method 200 further proceeds to step 220.

[0061] In step 220 the file is loaded in the content identification engine 120 and the file is processed. For this processing CPU time of the central infrastructure 100 is used. The content identification engine 120 can comprise, as described above, a conventional virus checker, a means for checking picture information, a means for checking SPAM, etc. This can be a repetitive action where multiple content identification engines are called in turn. Method 200 then proceeds to step 222.

[0062] In step 222 content attributes, which identify the content of the file, are determined for the file. These content attributes are then stored in the metabase 110, thus allowing to identify the status of the file if, in future operations, the file is found 'new' on another local computing device 50. Method 200 then proceeds to step 224. Depending on the embodiment used, a following step may include the storing of the file on the central infrastructure 100 and adding the path to this file to the metabase 110. This step is not shown in figure 3.

[0063] In step 224 the content attributes are sent to

Trend Micro, Inc.
EX1007-568

the local agent. Based on this content attributes, the local agent performs an appropriate action in agreement with the policy rule set for these content attributes. This is performed in step 226. This can be e.g. deleting the file if it was infected, replacing the file with a previous version which was not infected, ... In a specific embodiment, the execution of appropriate actions based on the policy rules are triggered by the agent of the metabase 110, so that step 224 can be avoided.

[0064] The content policy is a policy that determines what should be done with a file depending on the content attributes determined by the content identification engine 120. The content policy can comprise actions such as deleting the file, deleting the file and replacing it with a previous version, copying the file onto another computing device while leaving a copy on the originating computing device, moving the file onto another computing device while deleting the original file on the originating computing device, logging the presence of the file, changing the attributes of the file like hiding it or making it read-only, making the file unreadable, making the file un-executable, etc. The content policy will be executed by the local agent, e.g. when the content attributes are received from the central infrastructure 100. The content policy for that agent will be downloaded to the local computing device 50 by the agent from a central policy infrastructure.

[0065] Fig. 4 shows a method 300 of the content identification process triggered by the content identification engine 120 according to the above mentioned embodiments . The different steps that occur during the process, both on a local computing device 50 and on the central infrastructure 100 are discussed.

[0066] This process typically is used in situations where new algorithms or tools are being used for content identification. Such new algorithms or tools can either be optimised algorithms and tools or previously uninstalled tools. As mentioned earlier this may be regulated by a policy : the triggering of the content identification process may be determined by the type of new algorithms and tools that are being used for content identification.

[0067] Method 300 is initiated by change of the content identification engine 120, e.g. by providing new algorithms or tools for the content identification engine 120.

[0068] In the first step 302 of method 300 upon triggering, the metabase 110 is scanned for hash values corresponding with hashing keys. Method 300 then proceeds to step 304.

[0069] In step 304, a file that corresponds with the hashing key is requested. This file can be either requested from the central storage on the central infrastructure 100 or it can be requested from a local computing device 50. The local computing device 50 then gives permission to the central infrastructure 100 to upload the corresponding file. The path to the file corresponding with the hash value is available from the record corresponding with each hash value. If the record stores different paths all corresponding with a copy of the corresponding file, the agent on the central infrastructure 100 retrieves one copy of the file, e.g. by scanning the paths listed in the record until a local computing device 50 has been found that is at that time connected to the network 10 and that allows uploading of the file. Method 300 then proceeds to step 306.

[0070] Once the file has been retrieved, the file is sent to the content identification engine 120. This is performed in step 306. The upgraded content identification engine 120 then scans the content of the file and produces content attributes corresponding with the file. Method 300 further proceeds to step 308.

[0071] In step 308, the content attributes are stored in the metabase 110, to allow in future security steps to immediately identify the content of the files. Method 300 further proceeds to step 310.

[0072] In step 310, the content attributes are sent to every local agent that resides on a local computing device 50 whereon the corresponding file is stored. The paths can be found in the record of the corresponding hashing key stored in the metabase 110. In this step, content attributes are send to every file for which a path is mentioned in the record of the corresponding hashing key. Method 300 proceeds to step 312.

[0073] In step 312, the local agent on the corresponding local computing devices 50 executes the policy according to the content attributes and the according to the local computing device 50.

[0074] One of the major advantages of the embodiments of the invention is that a file new to the entire network 10 only needs to be scanned once. If on another local computing device 50, an identical copy of this file is used, installed, opened or saved and closed, the file will be recognised by the central infrastructure 100 as being known to the network 10, in this way avoiding the need to re-check the content of the file. This especially is advantageous if the invention is used for networks 10 having a large number of local computing devices 50.

[0075] When operating, the method and system for identifying the content of new files optionally can comprise checking 'the heartbeat' of the local agent at regular times, i.e. it can be checked whether the local agent is still running on the local computing device 50. This can avoid that a user locally shuts down the agent, thus making the local computing device 50 vulnerable. If the local agent has been shut down, the network administrator can be warned. Furthermore a warning message could be send to the local computing device 50 thereby warning the user of the local computing device 50. The network administrator could also put the local computing device 50 in quarantine so that it can not damage other local computing devices 50 in the network 10. Furthermore, the central agent can also try to rerun the local agent.

[0076] In a similar way, the method and system for identifying the content of new files optionally can check

Trend Micro, Inc.
EX1007-569

at regular times whether the local computing device 50 is still connected to the network 10. If the local computing device 50 is not connected to the network 10 anymore, the local agent may further operate, storing hashing keys of new files in a waiting list to be checked once the network connection is restored. In the mean time, the corresponding files may be put in quarantine or depending on the type of file e.g. may be prevented from being executed.

[0077] The above described embodiments may be used as a content firewall for the different computing devices connected to the external network. For every incoming/outgoing file, incoming/outgoing message or incoming/outgoing data frame, the content firewall calculates the hash, checks whether this is new, checks whether it is tagged for specific content and enforces the policy associated with the specific content.

[0078] In a further embodiment, another configuration for using the present invention as a content firewall is described. A schematic overview of a computer network wherein this method and system may be used, is shown in Fig. 5. Only one reconfigurable firewall electronic device 50, such as a local computing device which may be in the form of a dedicated reconfigurable firewall electronic device, is directly connected to an external network 400 such as e.g. the internet, and the remaining local computing devices 410, are not directly connected to the external network 400, but grouped in a network environment and only connected to the external network 400 by their connection to the electronic reconfigurable firewall device.. The external network may be any possible network available. It is a goal of the content firewall as represented by the reconfigurable electronic firewall device 50 to protect the network environment comprising the remaining local computing devices 410 from attacks originating from places and/or devices in the external network. The reconfigurable electronic firewall device 50 50 either contains a local copy of the metabase or it can use a high speed secured network to a central infrastructure 100 which is part of the internal network. This allows for fast queries through the metabase. During operation, the reconfigurable electronic firewall device 50 functioning as a content firewall performs the following actions : the hash value of incoming files or incoming messages or incoming data frames are calculated. Subsequently, the calculated hash values are compared with the metabase, which is either stored locally or by using a high speed secured network, and it is determined whether the incoming file, incoming message or incoming data frame is new. Furthermore, it is checked whether this file, this message or this data frame is tagged for specific content. Depending on the specific content, a policy is enforced which is associated with the specific content. This policy may be to let it pass through to its final destination, to drop it, to log it, to put it in quarantine, etc. This system requires sufficient CPU power, in order not to slow down the network speed noticeably.

[0079] In the case where none of the local computing devices connected to the network is equipped with a removable device, i.e. allowing for non-scanned content to be opened or executed on that device, this is a very secure and managable setup.

[0080] In another embodiment of the invention, a similar configuration for use of the present invention as a content firewall in promiscuous mode is provided. The content firewall thereby sees all traffic passing by, executes the hashing and comparing functions and contacts the agents to enforce a policy. The advantage of this approach is that there is no single point of failure and no bottleneck anymore and furthermore that still no resources are used on the local computing devices for calculating hashes. Furthermore, no bandwidth is used for contacting the central metabase. The disadvantage is that local agents need to be installed on all computing devices of the internal network.

In accordance with the above described embodiments, the present invention includes a computer program product which provides the functionality of any of the methods according to the present invention when executed on a computing device. Further, the present invention includes a data carrier such as a CD-ROM or a diskette which stores the computer product in a machine readable form and which executes at least one of the methods of the invention when executed on a computing device. Nowadays, such software is often offered on the Internet, hence the present invention includes transmitting the printing computer product according to the present invention over a local or wide area network.

Claims

1. A method for identifying the content of a file in a network environment, said network environment comprising at least one local computing device linked to a remaining part of the network environment including a central infrastructure and, the method comprising

   - calculating a reference value for a new file on one of said at least one local computing devices using a one-way-function,
   - transmitting said calculated reference value to said central infrastructure,
   - comparing said calculated reference value with reference values previously stored within the remaining part of the network environment,
   - after comparing,

     - deciding that the content of the new file is already identified if a match between said calculated reference value and a previously stored reference value is found and retrieving the corresponding content attributes; or

Trend Micro, Inc.
EX1007-570

- deciding that the content of the new file is not yet identified if no match between said calculated reference value and any of the previously stored reference values is found, followed by said central infrastructure identifying the content of said new file and determining content attributes corresponding with the content of the new file and storing a copy of said content attributes,
- after deciding, triggering an action on said local computing device in accordance with said content attributes.

2. A method according to claim 1, wherein the reference value is a hash value.

3. A method according to claim 1 or 2, wherein reference values previously stored are stored within the central infrastructure.

4. A method according to the previous claim wherein said identifying the content of said new file comprises scanning the new file for viruses using a anti-virus checker means on said central infrastructure.

5. A method according to any of the previous claims furthermore comprising transferring the new file from the local computing device to the central infrastructure before said identifying the content of said new file is performed.

6. A method according to claim 5, furthermore comprising storing a copy of the new file on the central infrastructure after said transferring the new file from the local computing device to the central infrastructure

7. A method according to any previous claim wherein said triggering an action on said local computing device in accordance with said content attributes may comprise replacement of the new file on the local computing device with a copy of a previous version of said new file.

8. A method according to any previous claim, wherein said triggering an action on said local computing device in accordance with said content attributes may comprise replacement of the new file on the local computing device with another version of said new file restored from the remaining part of the network environment.

9. A method according to claim 1, 2, 3 or 4 furthermore comprising sharing the new file on the local computing device to the central infrastructure before said identifying the content of said new file is performed and whereby said identifying the content of said new file is performed by remotely identifying the content over the network environment.

10. A method according to any of the previous claims, wherein said triggering an action on said local computing device in accordance with said content attributes is performed after transmitting the content attributes corresponding to the new file to the local computing device.

11. A method according to the previous claims wherein said identifying the content of said new file comprises one or more of the group of scanning for adult content, scanning for Self Promotional Advertising Messages and scanning for copyrighted information.

12. A method according to any of claims 1 to 3 or 5, whereby one local computing device links said network environment with an external network, said one local computing device acting as a content firewall as to protect the network environment.

13. A computer program product for executing any of the methods as claimed in claims 1 to 12 when executed on a network.

14. A system for identifying the content of a file in a network environment, said network environment comprising at least one local computing device linked to a remaining part the network environment which includes a central infrastructure and, said remaining part including a stored database, whereby the system comprises :

- means for calculating a reference value for a new file on said local computing device using a one-way-function,
- means for transmitting said calculated reference value to said central infrastructure,
- means for comparing said calculated reference value with previously stored reference values from the database,

whereby the system further comprises:

- means for deciding whether the content of the new file is already identified based on comparison of said calculated reference value and reference values previously stored within the remaining part,
- means located on the central infrastructure, for identifying the content of the new file and as to assign content attributes if the new file has not been identified yet and means for storing said content attributes within the remaining part, and
- means for triggering an action on said local

Trend Micro, Inc.
EX1007-571

computing device in accordance with content attributes for said new file.

15. A system according to claim 14 wherein said means for identifying the content of a file comprise an anti-virus checker means on said central infrastructure.

16. A system according to claim 14 or 15 furthermore comprising means for storing a copy of the new file within the remaining part.

17. A system according to any of claims 14 to 16 wherein said means for identifying the content of a file comprises one or more of the group of means for scanning for adult content, scanning for Self Promotional Advertising Messages and scanning for copyrighted information.

18. A method for altering a system for identifying the content of a file in a network environment according to claims 14 to 17, said network environment comprising means for calculating a one-way function, at least one local computing device linked to a remaining part of the network environment including a central infrastructure and means for identifying the content and, the method comprising

- altering said means for identifying the content or said means for calculating a one-way function
- scanning the remaining part of the network environment for reference values calculated with a one-way function
- for each of said reference values,
    - requesting a file that corresponds with said reference value from said network environment
    - sending the file to means for identifying the content
    - identifying the content of said file and determining content attributes corresponding with the content of the file and storing a copy of said content attributes
    - sending the content attributes to every local computing device containing the file
    - after sending; triggering an action on said local computing device in accordance with said content attributes.

19. A computer program product for executing the method as claimed in claims 18 when executed on a network.

20. A machine readable data storage device storing the computer program product of any of claims 13 or 19.

21. Transmission of the computer program product of any of claims 13 or 19 over a local or wide area telecommunications network.

Trend Micro, Inc.
EX1007-572

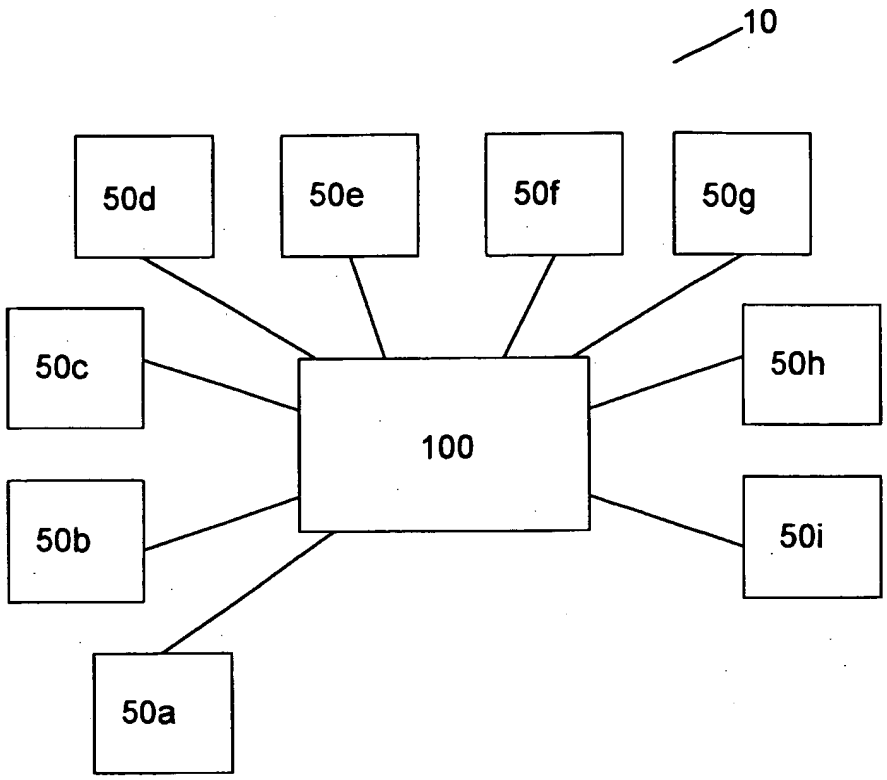


Fig. 1

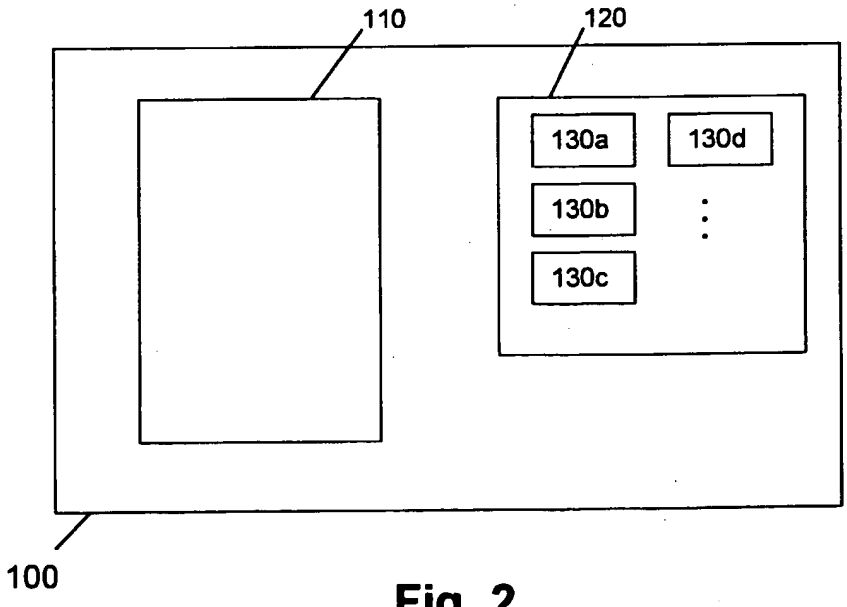


Fig. 2

Trend Micro, Inc.
EX1007-573

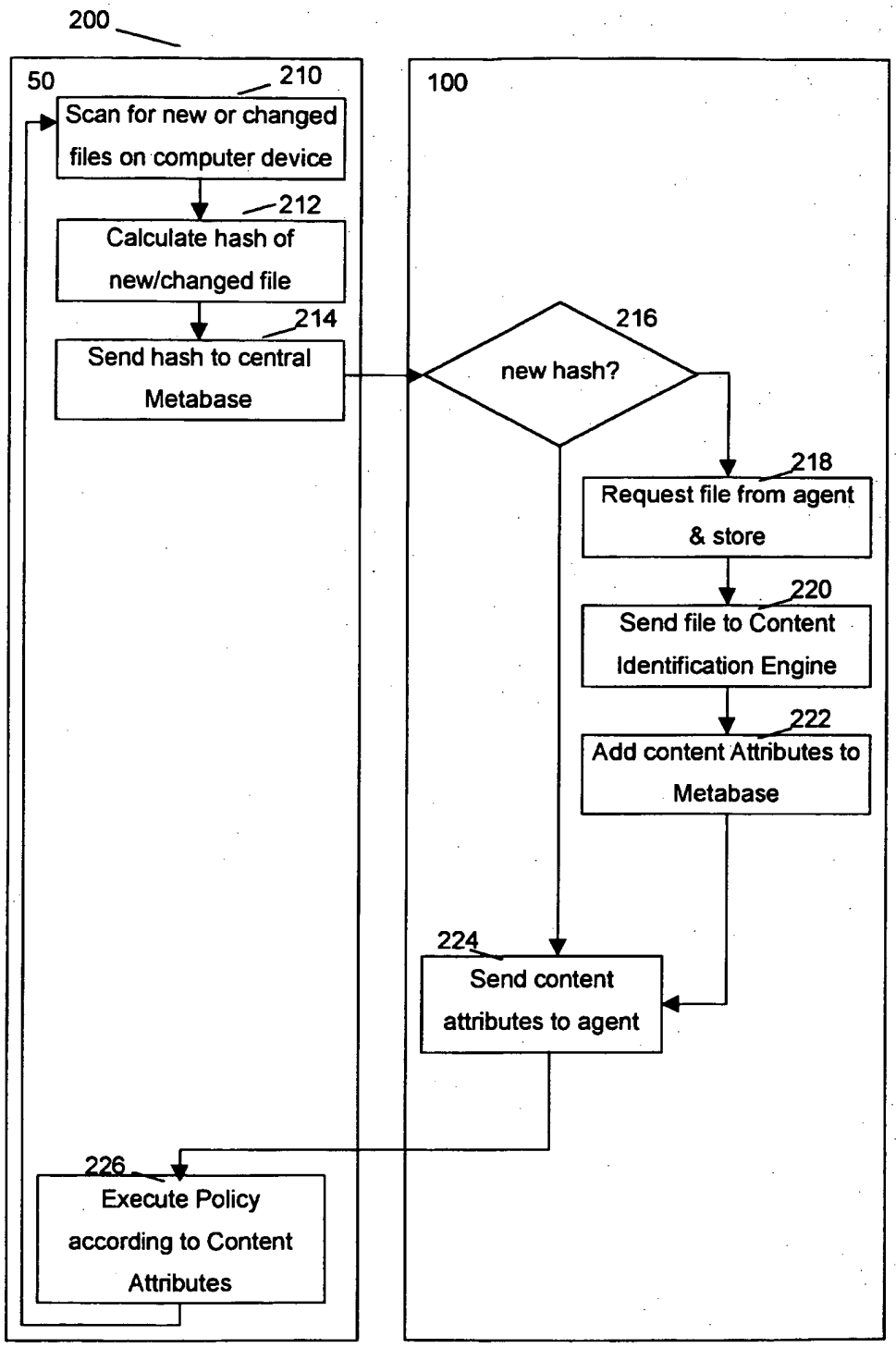


Fig. 3

Trend Micro, Inc.
EX1007-574

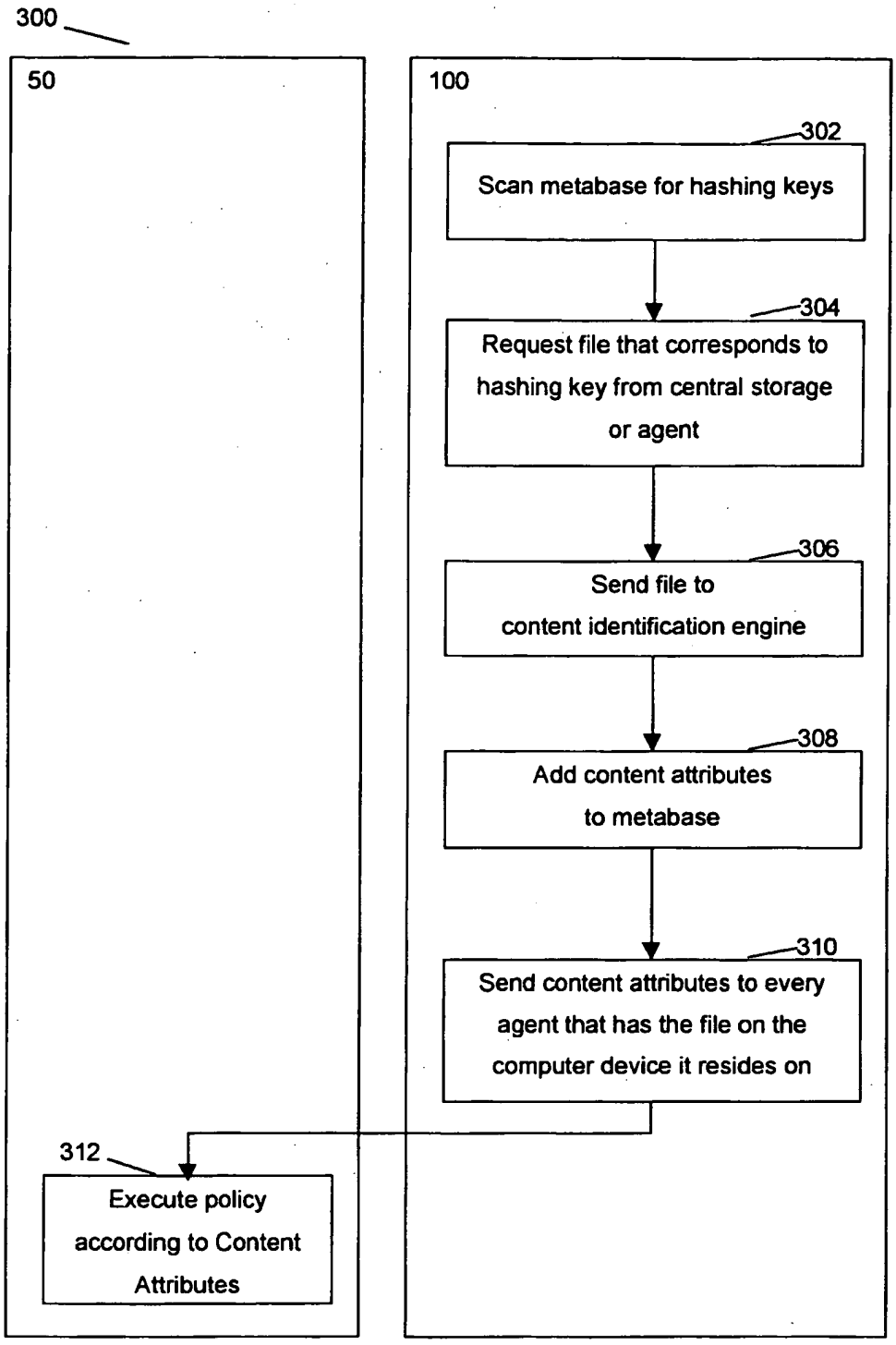


Fig. 4

Trend Micro, Inc.
EX1007-575

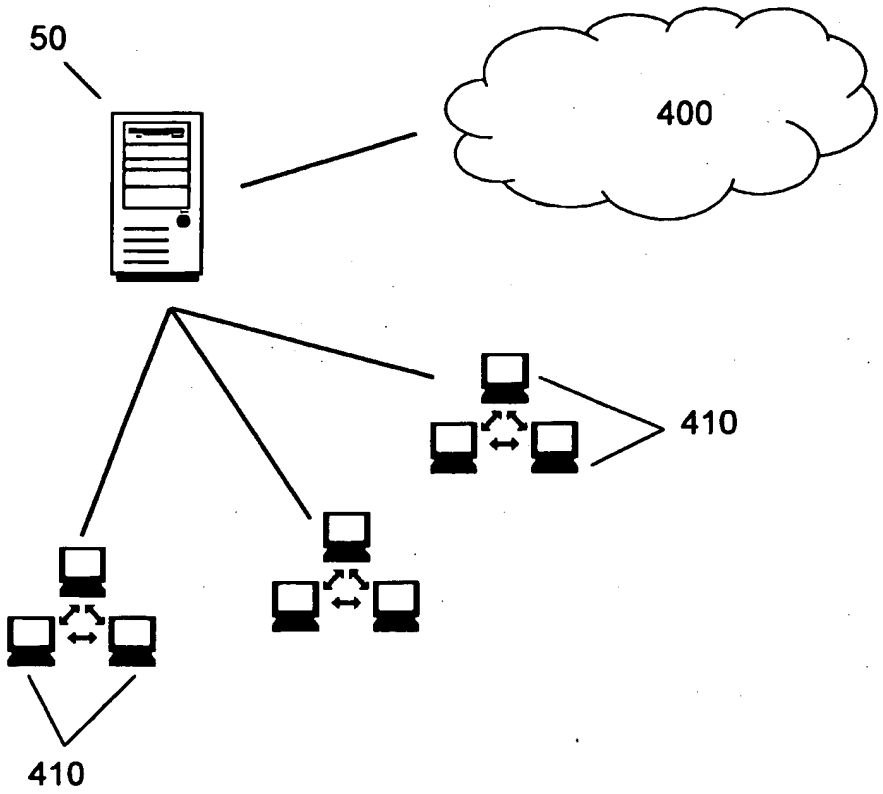


Fig. 5

Trend Micro, Inc.
EX1007-576

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 03 44 7310

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
X	EP 1 280 040 A (NETWORKS ASSOCIATES TECHNOLOGY) 29 January 2003 (2003-01-29) * the whole document *	1-9, 11-17, 20,21	H04L29/06 G06F1/00
A	---	10,18,19	
A	US 6 195 587 B1 (HRUSKA JAN ET AL) 27 February 2001 (2001-02-27) * column 1, line 4-9 * * column 1, line 50 - line 65 * * column 3, line 55 - column 5, line 67 * * column 1, line 50 - column 3, line 3 * * column 3, line 55 - column 5, line 40 *	1-6, 9-11, 13-20	
A	---		
A	US 2003/120953 A1 (RADATTI PETER V) 26 June 2003 (2003-06-26) * paragraphs [0013]-[0015] * * paragraph [0045] * * paragraph [0047] * * paragraph [0051] *	1-4,7,8, 11, 13-15, 17,20	
	---		TECHNICAL FIELDS SEARCHED (Int.Cl.7)
A	EP 1 111 507 A (SECURE OYJ F) 27 June 2001 (2001-06-27) * paragraphs [0002]-[0004] * * paragraphs [0010]-[0012] * * paragraph [0015] * * paragraphs [0025],[0026] * * paragraph [0031] *	1,4-8, 12-16,20	H04L G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	7 June 2004	Ruiz Sanchez, J

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

Trend Micro, Inc.
EX1007-577

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 03 44 7310

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

07-06-2004

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 1280040 A	29-01-2003	US 2003023866 A1	30-01-2003
		EP 1280040 A2	29-01-2003
US 6195587 B1	27-02-2001	GB 2283341 A	03-05-1995
		AU 8000394 A	22-05-1995
		DE 69407812 D1	12-02-1998
		DE 69407812 T2	16-07-1998
		EP 0725951 A1	14-08-1996
		WO 9512162 A1	04-05-1995
		JP 9504395 T	28-04-1997
US 2003120953 A1	26-06-2003	NONE	
EP 1111507 A	27-06-2001	GB 2353372 A	21-02-2001
		EP 1111507 A2	27-06-2001
		US 2001005889 A1	28-06-2001

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

Trend Micro, Inc.
EX1007-578

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
13 March 2003 (13.03.2003)

PCT

(10) International Publication Number
WO 03/021402 A2

(51) International Patent Classification[7]: G06F 1/00

(21) International Application Number: PCT/GB02/04059

(22) International Filing Date:
5 September 2002 (05.09.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
0121497.2 5 September 2001 (05.09.2001) GB

(71) Applicant *(for all designated States except US)*: **CRYPTIC SOFTWARE LIMITED** [GB/GB]; The Business Exchange, Rockingham Road, Kettering, Northants NN16 8JX (GB).

(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: **DUKE, David, John** [GB/GB]; 3 The Saplings, Kettering, Northants NN16 9UL (GB).

(74) Agent: **CRAWFORD, Andrew, B.**; AA Thornton & Co., 235 High Holborn, London WC1V 7LE (GB).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 03/021402 A2

(54) Title: NETWORK SECURITY

(57) Abstract: A computer network comprises a server computer and a plurality of local computers. Each local computer is provided with an engine program which monitors all files and peripherals and detects charges in the files or configuration. The results of the monitoring are used to create compressed information relating to the files and peripherals which is communicated to the server computer which stores previous information relating to the files and peripherals of the local computers. The server computer compares the received information with the stored information in order to detect the presence of altered and unwanted files or changes in configuration. In this way both known and unknown viruses can be detected as can un-authorised use of the local computers.

Trend Micro, Inc.
EX1007-579

NETWORK SECURITY

The present invention relates to an arrangement for monitoring the security of a computer especially computers on a network and particularly individual devices and information contained therein.

It is well known to provide individual computers with virus checking software but it is also well known that such software consumes resources when it is run and also takes time. Additionally, updating such software is somewhat cumbersome as each individual computer has to be updated separately.

Further, it is known to provide firewalls to protect computers and computer systems which are connected to a network or to the outside world but again these firewalls have limitations and they do not provide a completely effective defence to potential hackers.

Both of these methodologies are reactive and only detect threats to a network which have been seen and indexed previously.

It is an object of the present invention to provide an arrangement whereby protection from attack by viruses is improved as well as providing a more effective firewall in the event that the computer or computer system is provided with such. It is a further object that the invention is deployed in such a way that it is itself not vulnerable to attack.

From one aspect, the present invention is based on a concept whereby a main memory holds a record of all files used by a local computer, is provided with update information on all existing files as well as all new files and determines whether the updated files or new files represent a security threat.

In a preferred embodiment, as data is created, modified or deleted on the local computer, events are monitored and collected by an engine program and forwarded to the main memory. Preferably, the engine program itself consists of a

single file which is arranged to run a program in parallel with the normal operating system of the local computer thus making the file easier to hide so that the presence of the engine program will not be noticed either by the user or by a hacker. Further, as far as the user of the local computer is concerned, the engine program has no noticeable effect on the system. Preferably, the main memory is provided by a central computer (hereafter referred to as the console) and the local computer communicates with the console over a network or in some other convenient manner.

The engine program monitors each and every file and by inspecting preselected parts of each file can create a highly compressed accurate image of the file which can then be transmitted to the central database. It will be appreciated that all files can be handled in this way. Consequently, when the term "file" is used we mean all files associated with the operating system, data, registries, directories, hardware, software and such like. In this way, a complete virtual image of a local computer can be stored within the central database and the console can have management software for monitoring changes in the data files, programs or hardware of the local computer and thus warn of tampering with such files, programs and/or hardware.

From another aspect, the present invention provides a method of monitoring the security of a computer system comprising monitoring a file as it is created or updated by inspecting preselected portions of the file selected from the whole file, storing information derived from the preselected portions and transmitting the stored information to a main memory location. The main memory location is preferably a central database of a network.

From a further aspect the present invention provides a method of improving the security of computer apparatus by providing two copies of a security program and interconnecting them such that as one is switched off, either deliberately or

inadvertently, the other is automatically switched on and vice versa. It is to be understood that this aspect is not limited to any particular type of security program and is of general application. However, it has particular application with the present invention where the security program is relatively small and easy to hide.

In order that the present invention be more readily understood, an embodiment thereof will now be described with reference to the accompanying drawings in which: -

Fig 1 shows a block diagram of a network according to the present invention;

Fig 2 shows a block diagram representing the main functions of an engine program arranged to be installed on each computer to be protected; and

Fig 3 is a flow chart for explaining the operation of the engine program of Fig 1.

The present invention will be described in relation to its use for network security but it will be appreciated that the invention can be implemented in a number of different ways independently of any connection technology. In fact, the totality of the invention could be located within a single computer or a local computer could be in communication with a central server over conventional telephone lines rather than utilising any network protocol.

As shown in Fig 1, a central server 1 communicates via transmit/receive ports 2 with a number of local computers 3, each of which is provided with a monitoring engine program 4. The central server is provided with a plurality of databases, namely an archive of threats database 5, a virtual image of each computer database 6 and an audit database 7.

The basis of the present embodiment is that a virtual copy of a local computer and all its files will be kept in memory at the database to the central server computer with each file having its own individual characteristics stored at

the central computer. The virtual copies are created by the small engine programs 4 loaded on the local computers 3 which monitor all files by inspecting preselected portions of the files in order to create a so-called "fingerprint" of each file which is in fact a highly compressed version of the file. The fingerprint can then be very speedily forwarded to the central location where the current fingerprint can be compared with a previous fingerprint and any changes detected. The changes are then evaluated by the central computer 1 in order to determine the level of threat, if any, to the security of the local computer system and an appropriate signal sent back from the central server to the local computer 3 in the event that a particular file should not be opened or a particular program should not be run. Simultaneously, an event is displayed at the central computer and/or forwarded to other programs.

It will be appreciated that by the use of a virtual copy of the local computer and all its files, the central server can then monitor and log all changes to files, programs and/or hardware in order to provide evidence of breaches of security at the local computer. As an example of this, one could consider a computer fitted with several pieces of memory. Removal of all or part of this memory is an event which can be detected by the engine program and thus signalled to the central computer which can log the event.

In further explanation of the evidential facilities of this invention on the triggering of specified events, biometric information can be gathered in real time and forwarded to the central database. For example, if an event is detected, a screen shot of the local computer is gathered for evidence and at the same time the engine program can gather biometric information such as a photograph of the user of the local computer.

The above description represents a simple brief overall view of the arrangement in accordance with the present invention but for a more detailed

description we will now separate the operation of the engine program at the local computer and firstly describe this before we move onto a description of the operation of the monitoring program at the central server.

The engine program is shown diagrammatically in Fig 2 where all file data is monitored at monitoring section 11 under the control of a program section 12 which determines which parts of the data will be sampled. The selected parts of the data are then stored as indicated at 14 in order to create a virtual compressed copy of the file data for transmission as indicated at 16.

The engine program is arranged to commence running as soon as the computer on which it is loaded is powered up. The program then monitors all files as they are opened and in particular it monitors all changes to a file as indicated by the flow chart shown in Fig 3. Each and every change is inspected in order to determine whether the change is the creation of a new file. It also determines whether the change is a valid change. Each file is also checked by inspecting the file from a plurality of points of view by sampling predetermined portions of the file. This in turn creates a fingerprint or virtual copy of the file which is then assembled for transmission to a central database either over the network or via some other communications link. Among the predetermined portions of the file which the engine program inspects is the tag indicating whether or not the file is an executable. It also determines whether the file is a manipulated file, eg a zipped or encrypted file. It looks for the presence of capabilities such as keystroke logging, FTP server capability, IP notification, joystick controls, game libraries, etc

The engine program has a memory capability 12 so that it can store instructions received from the central processor as described below and then carry out those instructions in the event that the fingerprint it has assembled of a file indicates that action is necessary. As an example, if a known threat is present in a

file as indicated by the fingerprint, the engine program can immediately kill the threat in view of the fact that it has the necessary instruction previously received from the central server. In certain circumstances, the file in question can be automatically dealt with.

A particular feature of the engine program is that it does not wait to be polled by a central computer. Rather, it itself generates a message for transmission to the central computer. This has the advantage that the engine program is immune to attack from a hacker because there is no "listening" port waiting for an incoming transmission. It is not until the engine program has communicated with the central server that a two way communication is possible and only after the engine program's communication with the central server can the central server transmit modified information to the engine program. However, the monitoring and modification of the files on the local computer occur in real time while the local computer is in operation either by being started up or while the local computer is running programs. Further, the engine program can itself run other software in order to transmit and/or record data as a result of the detection of a particular event or change to a file. As an example of this, if the local computer is fitted with a web cam, should particular files be modified, the engine program can take a picture of the user of the machine at the appropriate time and correlate the event with the actual user as evidence for subsequent use.

In an especially secure system, it is possible to provide two identical engine programs which are interlinked with each other so that in the event that one is disabled for any reason, eg being switched off, the other is activated before the first is terminated and vice versa. This ensures security in the event that one of the engine programs is detected by a hacker or other person seeking to tamper with the local computer.

Turning now to the central server 1, this contains a reception port 2 for receiving data transmitted from the local computer. As fingerprints are stored, the central computer builds up a virtual picture of each local computer, its hardware, programs and files generally in its database 6. In addition, the central server maintains a database 5 of all known security threats and viruses. As fingerprints of files are received these are compared with the information in the threats database 5 so that any threat can be detected instantaneously. Additionally, because the central server maintains an up to date virtual copy of each local computer with which it is in communication, in the event that it identifies a security threat in relation to one computer 3, it can then inspect all other local computers for similar configurations and then be ready to communicate the necessary instruction to another local computer 3 as soon as the other local computer communicates with the central server for any reason. In this way, security threats and the mechanism of deleting files or repairing infected files can be undertaken much more quickly and in fact even before a local computer might have been subjected to attack or infection. In other words, when a new threat is identified, it is added to the server database so that the central processor can check its data relating to the local computers to which it is connected ot identify other infected systems without the need to communicate with them. Therefore even if a laptop is disconnected or machines are powered down, threats can still be identified and remedial action taken once such machines are back in communication with the central server.

In the same way, the central computer can monitor changes to hardware and software using the information in the database 7 and consequently know when a security breach might have taken place. In this case, the central computer can log the event in question or signal the local computer to take the necessary action to

provide evidence of the possible security breach, eg by taking a web cam picture as well as recording the user name and password.

Another sophisticated ability of the server is to construct a dynamic accurate focused detection formula which can be custom built to detect dynamic changing threats which avoid typical security fingerprinting techniques.

The central server can communicate back to the local computer in the event of a security breach or potential security breach or it can communicate with some other communications device such as a mobile telephone or personal digital assistant by sending an e-mail SMS or fax to indicate the existence of the security breach or potential security breach. In this way, users of portable computers could be warned of security breaches relating to their machine without the need to actually switch it on and establish communications with the central server.

CLAIMS:

1. A computer arrangement comprising a central processor, first memory means for storing an operating system program for the central processor, further memory means for storing data relating to all files used by the computer arrangement, storage means separate from the first storage memory means for storing an engine program for monitoring files used or created by the computer arrangement, and means for communicating the results of the monitoring to the further memory means.

2. A computer arrangement according to claim 1, wherein the engine program is arranged to inspect preselected parts of each file and create a compressed version of the file.

3. A computer arrangement according to claim 1, wherein the further memory means is located remote from the central processor and first memory means.

4. A computer arrangement according to claim 3, wherein the means for communicating results to the further memory means comprises a network connection.

5. A computer arrangement according to claim 1, wherein the storage means is arranged to store two engine programs each of which is capable of being switched off and which are interconnected such that as one is switched off, the other is switched on.

6. A computer arrangement according to claim 1, wherein the engine program is arranged to monitor peripheral devices connected to the arrangement in order to detect the presence of new devices and the absence of previously connected devices.

7. A computer arrangement according to claim 6, wherein the engine program is arranged to record data relating to any change in peripheral devices or files.

8. A computer arrangement according to claim 3, wherein the further memory means is located in a further computer which is arranged to analyse the results of the monitoring.

9. A computer arrangement according to claim 8, wherein the further computer is arranged to transmit instructions to the central processor depending on the results of the analysis.

1/2

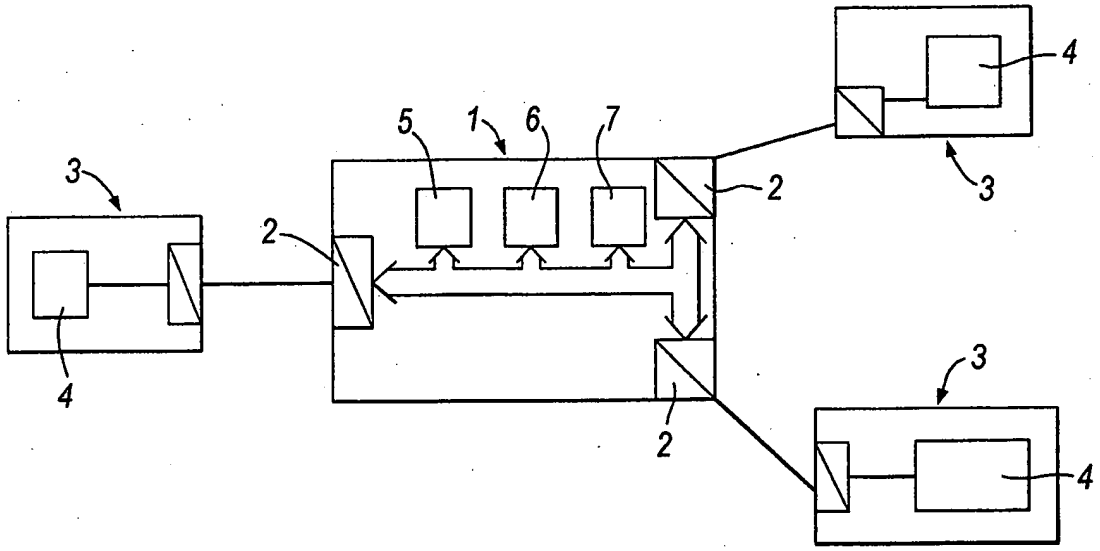


Fig.1

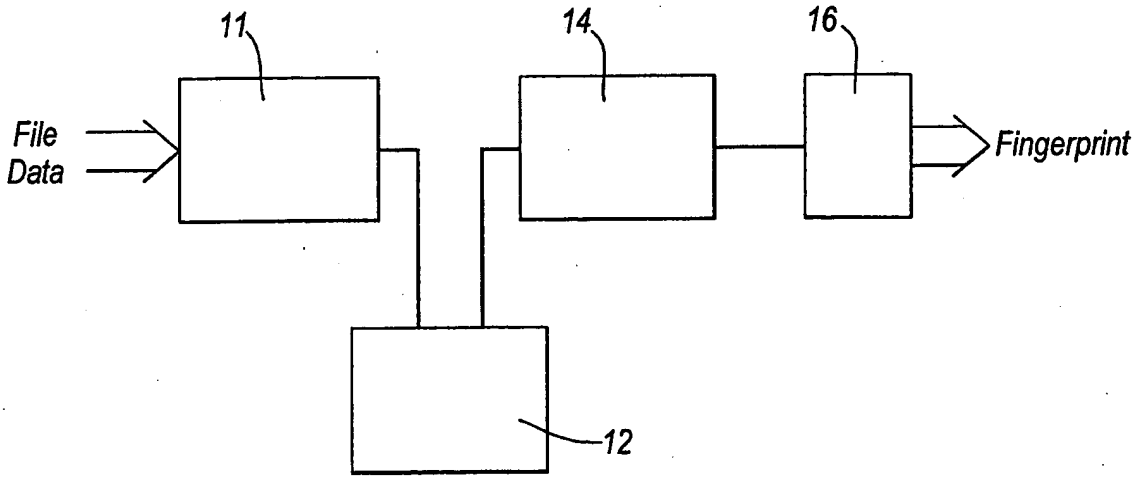


Fig.2

Trend Micro, Inc.
EX1007-590

2/2

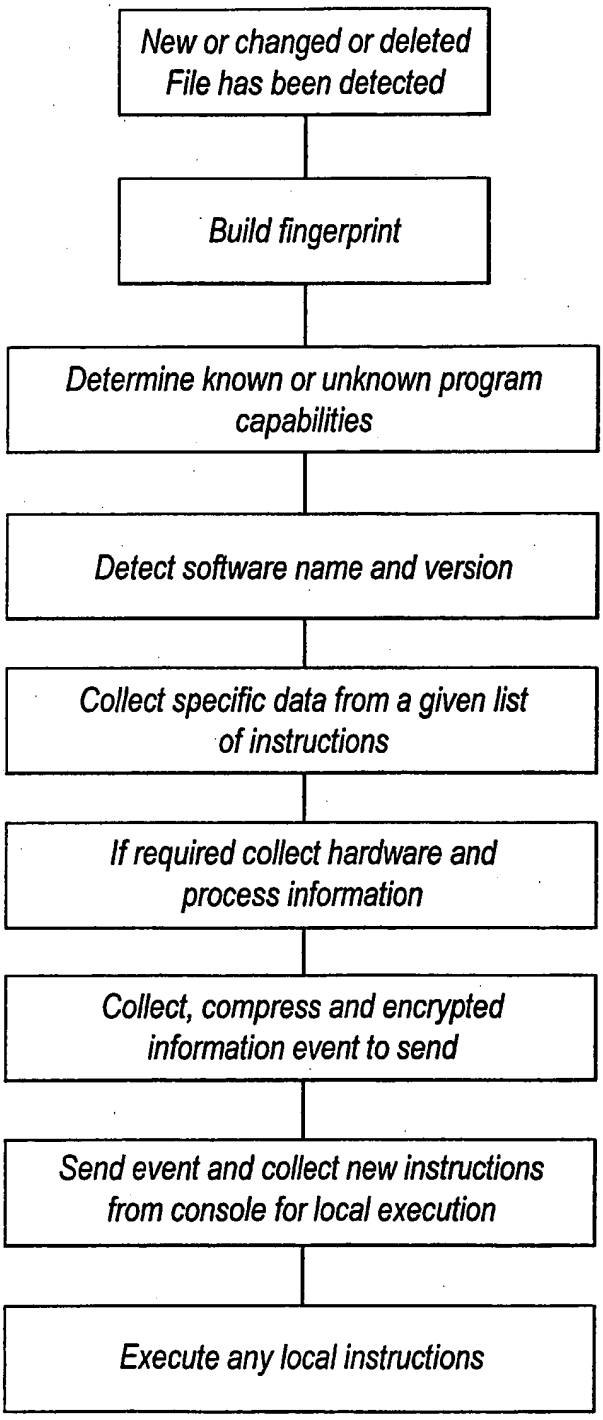


Fig.3

Trend Micro, Inc.
EX1007-591

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
13 March 2003 (13.03.2003)

PCT

(10) International Publication Number
WO 2003/021402 A3

(51) International Patent Classification[7]: G06F 1/00

(21) International Application Number:
PCT/GB2002/004059

(22) International Filing Date:
5 September 2002 (05.09.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
0121497.2 5 September 2001 (05.09.2001) GB

(71) Applicant *(for all designated States except US)*: **CRYPTIC SOFTWARE LIMITED** [GB/GB]; The Business Exchange, Rockingham Road, Kettering, Northants NN16 8JX (GB).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **DUKE, David, John** [GB/GB]; 3 The Saplings, Kettering, Northants NN16 9UL (GB).

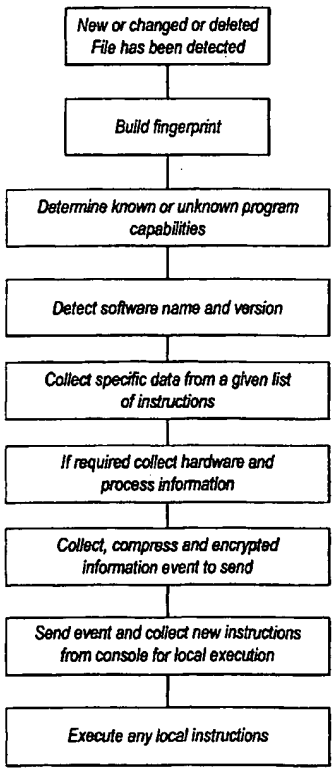
(74) Agent: **CRAWFORD, Andrew, B.**; AA Thornton & Co., 235 High Holborn, London WC1V 7LE (GB).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK,

[Continued on next page]

(54) Title: NETWORK SECURITY



(57) Abstract: A computer network comprises a server computer and a plurality of local computer. Each local computer is provided with an engine program which monitors all file and peripherals and detects changes in the files or configuration. The results f the monitoring are used to create compressed information relating to the files and peripherals which is communicated to the server computer which stores previous s information relating to the files and peripherals of the local computers. The server computer compares the received information with the stored information in order to detect the presence of altered and unwanted files or changes in configuration. In this way both known and unknown viruses can be detected as can un-authorised use of the local computers.

WO 2003/021402 A3

TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

(88) Date of publication of the international search report:
19 August 2004

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

INTERNATIONAL SEARCH REPORT

International Application No
PCT/GB 02/04059

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F1/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
WPI Data, EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 952 521 A (HEWLETT PACKARD CO) 27 October 1999 (1999-10-27) column 2, paragraphs 13,14 column 3, paragraphs 16,18,19 column 4, paragraph 22 -column 6, paragraph 29 column 10, paragraph 44 column 12, paragraphs 55,56 figure 1 ---	1-9
X	EP 0 899 662 A (HEWLETT PACKARD CO) 3 March 1999 (1999-03-03) column 4, paragraph 23 - paragraph 25 column 6, paragraph 31 -column 9, paragraph 43 figures 2,3 --- -/--	1-5,8,9

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
11 November 2003	04/12/2003

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Arbutina, L

Form PCT/ISA/210 (second sheet) (July 1992)

Trend Micro, Inc.
EX1007-594

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 475 625 A (GLASCHICK RAINER) 12 December 1995 (1995-12-12) column 2, line 10 - line 67 column 3, line 34 - line 46 column 5, line 12 - line 30 column 6, line 10 - line 18 column 8, line 17 - line 20 ---	1-4,8,9
X	GB 2 350 704 A (CARTER NICHOLAS PETER) 6 December 2000 (2000-12-06) page 2, line 3 - line 19 page 4, line 8 - line 10 page 7, line 1 - line 11 page 9, line 1 -page 10, line 2 page 25, line 12 -page 26, line 25 ---	1,3,4, 6-8
X,P	WO 02 33525 A (CHUANG SHYNE SONG) 25 April 2002 (2002-04-25) page 6, line 5 - line 25 page 7, line 15 - line 25 page 8, line 6 - line 25 figure 1 ---	1-4,8,9
X	WILLIAMS R N: "DATA INTEGRITY WITH VERACITY" INTERNET, 12 September 1994 (1994-09-12), XP002096828 Retrieved from the Internet: <URL:ftp://ftp.rocksoft.com/clients/rocksoft/papers/vercty10.ps> 'retrieved on 1999-03-16! abstract page 3, right-hand column, paragraph 3.1 page 8, left-hand column, paragraph 5.1 page 10, left-hand column, paragraph 5.8 ---	1
X	MCKOSKY R A ET AL: "A FILE INTEGRITY CHECKING SYSTEM TO DETECT AND RECOVER FROM PROGRAMMODIFICATION ATTACKS IN MULTI-USER COMPUTER SYSTEMS" COMPUTERS & SECURITY. INTERNATIONAL JOURNAL DEVOTED TO THE STUDY OF TECHNICAL AND FINANCIAL ASPECTS OF COMPUTER SECURITY, ELSEVIER SCIENCE PUBLISHERS. AMSTERDAM, NL, vol. 9, no. 5, 1 August 1990 (1990-08-01), pages 431-446, XP000147838 ISSN: 0167-4048 page 431, left-hand column, paragraph 1 -page 437, left-hand column, paragraph 4 figures 2,5 --- -/--	1

Trend Micro, Inc.
EX1007-595

INTERNATIONAL SEARCH REPORT

Int ional Application No
PCT/GB 02/04059

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
A	US 6 094 731 A (WALDIN RAY ET AL) 25 July 2000 (2000-07-25) column 1, line 59 - line 61	2

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

Trend Micro, Inc.
EX1007-596

INTERNATIONAL SEARCH REPORT

In ıal Application No
PCT/GB 02/04059

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0952521 A	27-10-1999	US 6282175 B1	28-08-2001
		DE 69911681 D1	06-11-2003
		EP 0952521 A2	27-10-1999
		JP 2000003347 A	07-01-2000
EP 0899662 A	03-03-1999	EP 0899662 A1	03-03-1999
		DE 69802294 D1	06-12-2001
		DE 69802294 T2	16-05-2002
		EP 1008048 A1	14-06-2000
		WO 9912098 A1	11-03-1999
US 5475625 A	12-12-1995	DE 4101141 C1	02-07-1992
		AT 112078 T	15-10-1994
		DK 567492 T3	09-01-1995
		WO 9213307 A1	06-08-1992
		EP 0567492 A1	03-11-1993
		ES 2059211 T3	01-11-1994
GB 2350704 A	06-12-2000	AU 4941300 A	28-12-2000
		WO 0075782 A1	14-12-2000
WO 0233525 A	25-04-2002	AU 9620501 A	29-04-2002
		WO 0233525 A2	25-04-2002
US 6094731 A	25-07-2000	US 6021510 A	01-02-2000
		CA 2351197 A1	18-05-2000
		EP 1129406 A1	05-09-2001
		WO 0028420 A1	18-05-2000
		AT 215714 T	15-04-2002
		CA 2311658 A1	03-06-1999
		DE 69804658 D1	08-05-2002
		DE 69804658 T2	14-08-2002
		EP 1032877 A1	06-09-2000
		WO 9927450 A1	03-06-1999

Trend Micro, Inc.
EX1007-597

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
11 November 2004 (11.11.2004)

PCT

(10) International Publication Number
WO 2004/097602 A2

(51) International Patent Classification[7]: G06F 1/00

(21) International Application Number:
PCT/GB2004/001333

(22) International Filing Date: 29 March 2004 (29.03.2004)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
0309463.8 25 April 2003 (25.04.2003) GB

(71) Applicant *(for all designated States except US)*: MESSAGELABS LIMITED [GB/GB]; 1270 Lansdowne Court, Gloucester Business Park, Gloucester LG3 4AB (GB).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: SHIPP, Alexander [GB/GB]; Star Internet, Brighouse Court, Barmwood, Gloucester GL4 3RT (GB).

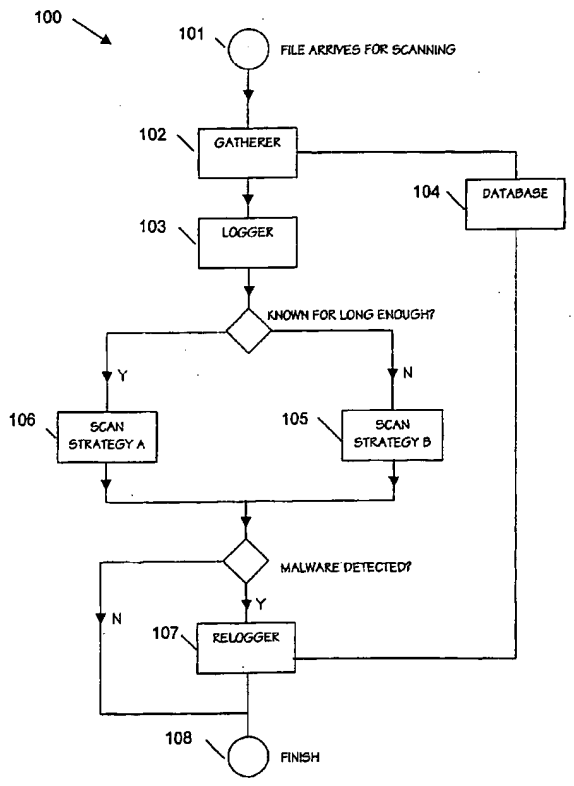
(74) Agents: AYERS, Martyn et al.; J.A. Kemp & Co., 14 South Square, Gray's Inn, London WC1R 5JJ (GB).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW),

[Continued on next page]

(54) Title: A METHOD OF, AND SYSTEM FOR, HEURISTICALLY DETERMINING THAT AN UNKNOWN FILE IS HARMLESS BY USING TRAFFIC HEURISTICS



(57) Abstract: A system for processing a computer file to determine whether it contains a virus or other maiware maintains a database of known files which it references to determine whether the file is an instance of a known file, and if so, whether it has been known about long enough that it can be regarded as safe. If it can be regarded as safe, the file is subject to less thorough processing for detecting malware, or no such processing at all.

WO 2004/097602 A2

Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

A METHOD OF, AND SYSTEM FOR, HEURISTICALLY DETERMINING THAT AN UNKNOWN FILE IS HARMLESS BY USING TRAFFIC HEURISTICS

The present invention relates to a method of, and system for, heuristically determining that an unknown file is harmless by using traffic heuristics. This technique is especially applicable to situations where files enter a system, are checked, then leave, such as email gateways or web proxies. However, it is not intended to be limited to those situations.

Increasing use of the Internet, personal computers and local- and wide-area networks has made the problem of viruses and other malware (=malicious software) ever more acute.

There are numerous anti-virus packages available. These tend to be produced by specialist companies and are used by businesses and other organisations, home users, and by some internet service providers (ISPs) who scan e-mail and other network traffic on behalf of their customers as a value-added service. As new viruses and other malware arise, the package creators devise ways of detecting them and dealing with them and issue updates to their packages which customers can utilise. A common practice is to make the updates available for download over the internet, from the creator's website or ftp site.

Most anti-virus packages include a file-scanning engine and a database of characteristics of known viruses which are used by the scanning engine to determine whether a file being scanned is, or contains, a virus or other malware, or is likely to do so. The sort of update mentioned above typically includes an update to this database.

The scanning engine may implement a variety of heuristics to be applied, possibly selectively, to a file being scanned. Probably the most familiar kind of heuristic is signature detection, in which the file is examined for the occurrence of sequences or bytes, or patterns of such sequences, which are known to be characteristic of viruses in the package's virus database, though many other heuristics also exist, which can be used as well as or instead of signature detection.

The amount of malware in existence increases all the time, which makes the computational and storage resources necessary to detect it increasingly burdensome, particularly where the throughput of files is high, as is the case with ISPs.

According to the present invention, there is provided a system for processing a computer file to determine whether it contains a virus or other malware comprising:

a) means for generating data with regard to the file to characterise its identity and for thereby referencing a computer database to determine whether it is an instance of a known file;

b) means for selectively subjecting the file to a number of heuristic procedures to determine whether or not it contains, or is likely to contain, malware; and

c) means for determining, in dependence upon the record, if any, of the file in the database, whether the file can be regarded as safe and for controlling the means b) such that the file, if the file is to be regarded as safe, is either subject to less thorough processing than if it were not so regarded or not subject to processing by the means b) at all.

The invention also provides a method of processing a computer file to determine whether it contains a virus or other malware comprising:

a) generating data with regard to the file to characterize its identity and for thereby referencing a computer database to determine whether it is an instance of a known file;

b) selectively subjecting the file to a number of heuristic procedures to determine whether or not it contains, or is likely to contain, malware; and

c) determining, in dependence upon the record, if any, of the file in the database, whether the file can be regarded as safe and conducting the step b) such that the file, if the file is to be regarded as safe, is either subject to less thorough processing than if it were not so regarded or not subject to processing by the step b) at all.

The invention will be further described by way of non-limitative example with reference to the accompanying drawings, in which:

Figure 1 is a block diagram of a system embodying the present invention.

Figure 1 illustrates one form of a system 100 according to the present invention, which might be used, for example by an ISP as part of a larger anti-virus scanning system which employs additional scanning methods on files which are not filtered out as "safe" by the system of Figure 1. Files considered safe can if desired be subject to further processing to check for malware, but less intensively so than files not considered safe.

The rationale of the system 100 is that if a particular file has been scanned by a virus scanner, and found to be harmless the two possibilities exist: The file could really be harmless, or the file could contain something nasty which the virus scanner is as yet unable to detect.

As time goes by, the file (or another instance of it) may be scanned again, and still found to be harmless.

This time the file is more likely to really be harmless, rather than to be malware which the virus scanner is as yet unable to detect. This is because virus scanners are continually updated to detect new malware as the new malware is discovered. The longer the time that passes, the more likely it is that a suspicious person will submit a file containing malware to the developers of the scanner, who will analyse the file, and update their scanner to detect it.

As more and more instances of the file are scanned coming from different sources, then if these are all flagged as harmless, it becomes less and less likely the file is malware. This is because the more copies of a piece of malware exist, the more likely it is that somebody will become suspicious and submit a copy to scanner developers.

It is therefore possible to create a feedback engine which logs copies of files scanned, together with the source they originated from. The log is updated and examined as each file is scanned, and if files are found which have come from a sufficient number of sources, in sufficient quantities, and over a long enough period of time, then that file can be flagged as 'known about long enough'. This might mean that future copies are then not scanned further, or are scanned using less rigorous scans with fewer heuristics enabled, or are only scanned if the scanner has been updated since the last scan.

The system 100 operates according to the following algorithm:

1) A file arrives at an input 101 for scanning, perhaps as an email attachment, or a web download.

2) A 'gatherer' module 102 gathers information about the file, such as a checksum of the file contents and the source of the file (eg the IP address). The source may be passed through a one way trapdoor function, generating a hash, in order to preserve confidentiality. The information gathered is for comparison with information stored in a database 104 about known files so that it can be determined whether the file under consideration is an instance of a file recorded in database 104.

3) Based on the checksum derived by gatherer 102, a 'logger' module 103 updates the database 104 to indicate that one more instance of the file has been detected.

The logger 103 saves the current 'last seen' date as the 'previously scanned date', and then updates the 'last seen' date of the file's entry in the database 104. If this is the first instance of the file, the logger 103 also updates a 'first seen' date. If this is a new source, the logger 103 adds the source to a list, stored in database 104, of sources the file has originated from.

4) From the information stored (number of copies of the file seen, length of time file has been known about, number of sources) the logger 103 calculates whether the file has been 'known about long enough'. For this purpose, the logger 103 may assign a weighted score to each of these factors individually and then calculate an overall score by combining the weighted scores, e.g. by adding them up.

5) If the file has not been known about long enough, scan strategy B is undertaken at 105. This will be the most complete scan available.

6) If the file has been known about long enough, scan strategy A is undertaken at 106. This will be a less thorough scan than strategy B. This will be site-dependent as to how less thorough a scan is desired. At the extreme it might involve no scanning at all. It might involve scanning with fewer scanners; with heuristics not fully enabled or turned off; or (assuming the file has been seen at least once before) only with scanners that have been updated since the 'previously scanned date'

The scanning techniques available to the scanning strategies A and B may include any suitable heuristics, such as signature-based scanning, generating checksums from the file or selected regions if it, etc.

7) Following the scan strategy A or B, then if no malware was detected, processing stops at 108.

8) If malware was detected, then a 'relogger' module 107 is invoked. This clears out all database entries in database 104 which are associated with the file so that it cannot become 'known about long enough' in the future.

9) Processing of the current file finishes at 108, whereupon the system can retrieve the next file from a queue of files waiting to be processed.

CLAIMS

1. A system for processing a computer file to determine whether it contains a virus or other malware comprising:

a) means for generating data with regard to the file to characterise its identity and for thereby referencing a computer database to determine whether it is an instance of a known file;

b) means for selectively subjecting the file to a number of heuristic procedures to determine whether or not it contains, or is likely to contain, malware; and

c) means for determining, in dependence upon the record, if any, of the file in the database, whether the file can be regarded as safe and for controlling the means b) such that the file, if the file is to be regarded as safe, is either subject to less thorough processing than if it were not so regarded or not subject to processing by the means b) at all.

2. A system according to claim 1 wherein the controlling means c) controls the means b) in dependence on factors including the length of time for which the database indicates that the file has been known without malware-containing instances of it being detected.

3. A system according to claim 1 or 2 wherein the controlling means c) controls the means b) in dependence on factors including sources, recorded in the database, from which instances of the file have originated.

4. A system according to claim 1, 2 or 3 wherein the controlling means c) controls the means b) in dependence on factors including the number of times, recorded in the database, of instances of the file have been processed.

5. A system according to any one of the preceding claims, and including means for updating the database in dependence upon the result of the processing of the file by the means b).

6. A system according to claim 5 wherein the updating of the database, in the event of the means b) determining that the file contains, or is likely to contain, malware is such that the record thereof in the database is deleted, or updated so that it is no longer taken be safe.

7. A method of processing a computer file to determine whether it contains a virus or other malware comprising:

a) generating data with regard to the file to characterise its identity and for thereby referencing a computer database to determine whether it is an instance of a known file;

b) selectively subjecting the file to a number of heuristic procedures to determine whether or not it contains, or is likely to contain, malware; and

c) determining, in dependence upon the record, if any, of the file in the database, whether the file can be regarded as safe and conducting the step b) such that the file, if the file is to be regarded as safe, is either subject to less thorough processing than if it were not so regarded or not subject to processing by the step b) at all.

8. A method according to claim 7 wherein the determining step c) controls the step b) in dependence on factors including the length of time for which the database indicates that the file has been known without malware-containing instances of it being detected.

9. A method according to claim 7 or 8 wherein the determining step c) controls the step b) in dependence on factors including sources, recorded in the database, from which instances of the file have originated.

10. A method according to claim 7, 8 or 9 wherein the determining step c) controls the step b) in dependence on factors including the number of times, recorded in the database, instances of the file have been processed.

11. A method according to any one claims 7 to 10, and including the step of updating the database in dependence upon the result of the processing of the file by the step b).

12. A method according to claim 11 wherein the updating of the database, in the event of the step b) determining that the file contains, or is likely to contain, malware is such that the record thereof in the database is deleted, or updated so that it is no longer taken be safe.

13. A system for processing a computer file to determine whether it contains a virus or other malware substantially as hereinbefore described and with reference to the accompanying drawings

14. A method of processing a computer file to determine whether it contains a virus or other malware substantially as hereinbefore described and with reference to the accompanying drawings

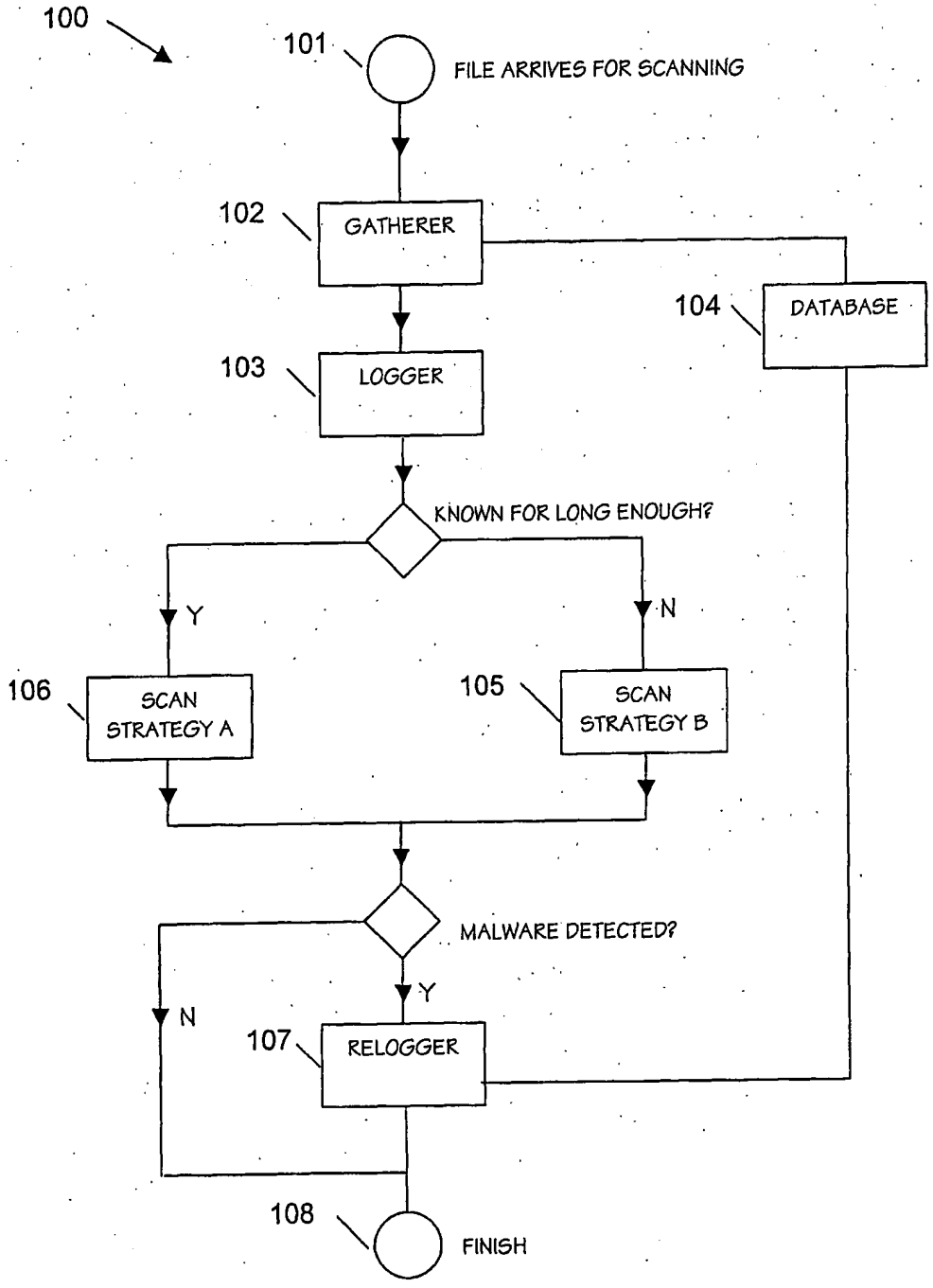

100
101
FILE ARRIVES FOR SCANNING
102
GATHERER
104
DATABASE
103
LOGGER
KNOWN FOR LONG ENOUGH?
Y
N
106
SCAN STRATEGY A
105
SCAN STRATEGY B
MALWARE DETECTED?
Y
N
107
RELOGGER
108
FINISH

Fig.1

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
25 April 2002 (25.04.2002)

PCT

(10) International Publication Number
WO 02/33525 A2

(51) International Patent Classification[7]: G06F 1/00

(21) International Application Number: PCT/SG01/00213

(22) International Filing Date: 17 October 2001 (17.10.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
200005973-3 17 October 2000 (17.10.2000) SG

(71) Applicant and
(72) Inventor: CHUANG, Shyue-Song [SG/SG]; Singapore Post Centre, P.O. Box 493, Singapore 914017 (SG).

(74) Agent: DREW & NAPIER LLC; 20 Raffles Place, #17-00 Ocean Towers, Singapore 048620 (SG).

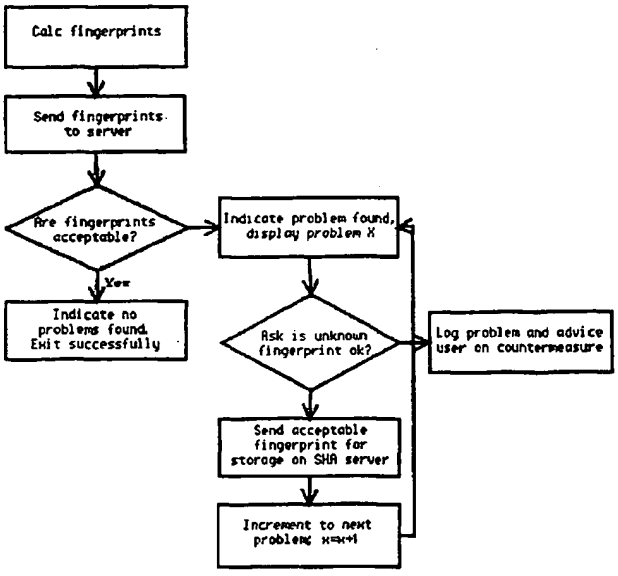
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: A METHOD AND SYSTEM FOR DETECTING ROGUE SOFTWARE



(57) Abstract: A method of detecting rogue software includes the step of creating a first database containing pre-calculated fingerprints for each file relating to typical operating systems and application software, wherein the pre-calculated fingerprints are calculated using one or more cryptographic formulae. The one or more cryptographic formulae are then used to calculate fingerprints of files on a computer system which is to be scanned for rogue software. The fingerprints calculated for the files on the computer system are compared with the fingerprints which are contained in the first database of pre-calculated fingerprints. Files on the computer system which may contain rogue software are identified by identifying files the calculated fingerprints of which do not correspond to the pre-calculated fingerprints which are stored in the first database.

WO 02/33525 A2

A METHOD AND SYSTEM FOR DETECTING ROGUE SOFTWARE

Field of the invention

The present invention relates to a method and system for detecting rogue software such as trojan horses, root-kits, viruses and other unauthorized software which masquerades as valid software) on a computer system or data processing device such as a personal digital assistant. It relates particularly but not exclusively to a method and system for calculating and comparing fingerprints for files which are used either on a stand-alone computer system or on a computer system which is part of a computer network.

Back-ground to the Invention

Undesired rogue software is a nuisance and security threat. As computer systems and other information devices become even more interconnected with modern day networking technology and the Internet, the danger from rogue software has magnified considerably. Instead of being programmed to do damage once, today's rogue software can continue to receive commands and do the bidding of an unauthorized intruder for an extended period of time, effectively giving the creator of the rogue software continuous illegal access to a computer system.

One example of rogue software is the so-called trojan horse. Such software may be installed by innocent users unknowingly (whether via social engineering or otherwise) or it may be installed by an attacker when a system has been broken into. These trojan horses are back doors which allow an attacker to reconnect back into the compromised system and illegally access files and make unauthorized changes.

A trojan horse typically consists of new software and has new functionality. It is installed on a compromised system and disguised to look like original system software whenever necessary, so as to avoid detection. Sometimes, the trojan

horse is a modified piece of original system software and is almost identical to the one it replaced. However, other techniques are also used to obfuscate its existence.

Once the trojan horse is installed, a user continues operating his/her system without knowing that an intruder is now able to access his/her data illegally whilst remaining hidden. These are highly relevant problems which are encountered on a day to day basis. Trojan horses such as "Back Orifice" or "Netbus" hit PC systems in the late 1990s, and "root-kits" are a concern for various UNIX systems.

Normally, according to current practice, little is done to prevent or detect such rogue software. Anti-virus vendors maintain databases of rogue software signatures, and their software searches for files on a system associated with all known rogue software. Unfortunately, this technique has inherent scaling problems - the more signatures there are, the slower the scan process for each file. More importantly, the only rogue software types that can be detected are the ones that the anti-virus vendors know about. If the rogue software is, as an example, a custom trojan horse built by an expert professional hacker for penetrating a specific target, none of the anti-virus vendors will know about it. Therefore none of the anti-virus tools will be looking for it and the attacker and their trojan horse will exist completely undetected.

A more recent incremental innovation with this technology involves smarter scanning engines. Aside from looking for signatures of known rogue software, they are also able to look for software code that appears to be doing unusual things. This allows the scanning engine to detect additional rogue software that may not be known and whose signatures may not be in the database. However, this approach also has limitations. Trojan horses can be encrypted or compressed using special proprietary algorithms or encryption keying material. The rogue software is shipped in an encrypted and/or compressed format where it appears to be gibberish to a scanner. This rogue software is then decompressed or decrypted upon execution on the victim's computer system. A single trojan horse can thus be encrypted or compressed into thousands of

possibilities, each with its own unique signature. Traditional scanning technology will fail miserably when attempting to detect this type of rogue software, since there is no way that anti-virus engineers can keep track of thousands of mutations of the same piece of rogue software.

To summarise, today's scanning techniques for detecting rogue software fail for two main reasons:

1. They cannot detect unknown rogue software that has not already been identified. This is a serious problem because it is this kind of rogue software that may involve professional hackers and therefore warrant serious attention.
2. They cannot efficiently detect rogue software which has mutated, using new methods of compression and/or encryption. This problem exists to a large extent even for rogue software that is already known.

Another approach for detecting rogue software is to ensure that a system's files have not been altered, rather than looking for signatures of rogue software. If a system has no added files and all files remain unchanged from their original, unaltered state, it is clear that no rogue software is present on the system.

Academic work at Purdue University by Gene Kim and Gene Spafford resulted in a product called Tripwire which is now commercially sold. The product requires users to generate a database containing fingerprints of files on a system when the system is freshly loaded and in a pristine state. Subsequently, fingerprints can be recalculated and compared with the database of original pristine fingerprints and detect changes which have been made to the computer system.

This technology requires users to generate a database of fingerprints of a system's files while it is still pristine and free from alteration. This is not always feasible because many systems would already have been placed on public networks and exposed to risk for some time (often years). Since changes can be detected only by calculating new fingerprints and comparing them with the

database of original fingerprints, any rogue software which already exists when the original fingerprint database was generated will not be detected.

Existing products do not use a central database of fingerprints which are acceptable for a broad collection of system and application software. Therefore users need to make the following tedious and expensive steps when installing a software upgrade:

1. Schedule downtime in which to create the new database of fingerprints;
2. Re-calculate fingerprints to ensure that no rogue software has been added;
3. Install the software upgrade; and
4. Generate a new fingerprint database.

This is often a time-consuming and costly exercise.

Summary Of The Invention

Therefore it is an object of the present invention to provide a more reliable and effective method of identifying rogue software on a computer system or device, especially rogue software with unknown signatures or characteristics. The invention is preferably usable on systems or devices that have already been exposed to risk of intrusion by rogue software, and in cases where no fingerprints for the files on the system were calculated or archived when the system or device was known to be in a pristine state.

According to a first aspect of the invention, there is provided a method for detecting rogue software including the steps of:

(a) creating a first database containing pre-calculated fingerprints for each file relating to typical operating systems and application software, wherein the pre-calculated fingerprints are calculated using one or more cryptographic formulae;

(b) using the one or more cryptographic formulae to calculate fingerprints of files on a computer system which is to be scanned for rogue software;

(c) comparing the fingerprints calculated for the files on the computer system with the fingerprints which are contained in the first database of pre-calculated fingerprints; and

(d) identifying files on the computer system which may contain rogue software by identifying files the calculated fingerprints of which do not correspond to the pre-calculated fingerprints which are stored in the first database.

According to a second aspect of the invention, there is provided a system for detecting rogue software including:

(a) a first database containing pre-calculated fingerprints for each file relating to typical operating systems and application software, the fingerprints having been calculated using one or more cryptographic formulae;

(b) a software component which uses one or more cryptographic formulae to calculate fingerprints for files on a computer system; and

(c) a software component which compares the calculated fingerprints for the files on the computer system with corresponding pre-calculated fingerprints stored in the first database, such that files on the computer system which may contain rogue software are identified by identifying files the calculated fingerprints of which do not correspond to the pre-calculated fingerprints which are stored in the first database.

Brief Description Of The Drawings

The invention will now be described in greater detail by reference to the drawings which show an example form of the invention. It is to be understood that the particularity of the drawings does not supersede the generality of the foregoing description of the invention.

Figure 1 is a schematic representation of a client portion and server portion of a security system on a Redhat Linux platform connected via a network according to a preferred embodiment of the present invention.

Figure 2 illustrates a more detailed data flow diagram relating to the schematic representation of Figure 1.

Detailed Description Of A Preferred Embodiment Of The Invention

Figure 1 is a schematic representation of a client portion and server portion of a security system on a Redhat Linux platform connected via a network 10 according to a preferred embodiment of the present invention. The system includes a client 12, a server 14 and a database of acceptable file fingerprints 16. Communication between the client 12 and server 14 may be via the Internet 18, using the TCP/IP protocol. The system is first set up by calculating and archiving fingerprints for all files relating to operating system or application software used in a typical Redhat Linux system, perhaps from original Redhat CDs or other secure software distribution methods. This software can be installed on test systems (not shown in Figure 1) so that the new files added or replaced can be fingerprinted and profiled. These new fingerprints, the file location of each file added or replaced, and other information, can then be stored in the database of acceptable file fingerprints 16. An alternative method that eradicates the need for software installation on a test system involves the use of a custom developed program that understands the RPM (Redhat Package Manager) format of software packages on the Redhat Linux CD. By examining each RPM software package's installation instructions, the program determines the file location and calculates the fingerprints of each file to be installed. This information and other information is then stored in the database of acceptable file fingerprints 16.

The fingerprints are preferably calculated using one or more cryptographic formulae. In the preferred embodiment, such cryptographic formulae may include hash functions to generate hash values for each file, or asymmetric cryptographic functions to generate digital signatures for each file. The original version of the files as well as patches, updates/upgrades of all types of operating system or application software should be fingerprinted. System performance and reliability will improve as more operating system and application software is fingerprinted and archived.

Hashing is a contraction of the file contents created by a cryptographic hash function. A hash value (or simply hash) is the output when an arbitrary input is passed into a hash function. The hash is substantially smaller than the input itself, and is generated by a formula in such a way that it is extremely unlikely that slight modifications of the input will result in the same hash. Hashes conventionally play a role in security systems where they are used to ensure that transmitted messages have not been tampered with. As an illustration, a sender generates a hash of the message and sends it with the message itself. The recipient then calculates another hash from the received message, and compares the two hashes. If they are the same, there is a very high probability that the message was transmitted intact. There may be other equivalent methods for calculating fingerprints that may be implemented as the relevant technology develops.

The system's client component is installed on the client 12 that requires file integrity protection. During the first time the client component is executed, the client software recurses through the file system and calculates and stores the cryptographic hash of every single file on the system. When the file system has been completely traversed, the client software makes a secure TCP/IP connection via the Internet 18 to the server component on the server, which usually resides on premises remote from the client component. However, the client component need not be physically located remote from the server component.

For security purposes, it is preferable that bi-directional authentication takes place between the client component and the server component before any further communication and this can be done with SSL (Secure Socket Layer) or TLS (Transport Layer Security). In a nutshell, the server presents its digital certificate to the client software and the client uses its hardwired CA (Certificate Authority) public credentials to verify the CA signature on the server's certificate. If the signature is authentic and the server's address matches the machine which the certificate was issued to, the client can be certain that the server is who it claims to be. Subsequently, the same thing happens in the reverse

direction. The client presents the server with its digital certificate and the server goes through the same process to verify that the client is who it claims to be. This practice is very common today and is an industry standard method of mutually authenticating two nodes communicating with one another. Other authentication methods may also be used.

The calculated hash results and gathered basic client system information from the client 12 are then transferred to the server 14 for validation. On the server 14, each hash result for each file on the client system is compared against what are the expected hash values given certain parameters such as the client system's operating system version and software patch/update level. This expected hash information is fetched from the database of acceptable file fingerprints 16 which houses all the pre-calculated hash values for all files in various operating systems and applications.

A report is then generated on the fly and returned to the client 12. This report lists the files on the client which are possibly unsafe since they do not represent authentic software from the vendor. There are 3 possible results for each file:

(a) the hash result matches so the given file on the client is definitely authentic;

(b) the database of acceptable fingerprints 16 has no information on such a file in the database and it is uncertain if the file is authentic;

(c) the hash result does not match the fingerprint in the database of acceptable fingerprints 16 and the file is suspicious.

Armed with this report, the systems administrator for the client server 12 can then verify each of the files in categories (b) and (c). Outcomes in categories (b) and (c) are typically from files that are part of an internal customer specific application that the database 16 will not contain. If the administrator verifies the hash with the owners of the application, the authenticity of the file can be determined. This should be done for all questionable files in the report so that a client system can be certified as 100% authentic. If some of the questionable files cannot be resolved via these means, it is likely that they have been

augmented by rogue software and should be replaced or the system should be reinstalled.

Using additional management software, the administrator can then check off all remaining questionable files as acceptable and the security system will take the additional hashes into account in all subsequent runs. These additional hashes can then be stored in a second database (not shown in Figure 1) so that they can be considered when checking other systems from the same customer - this is a configurable feature.

Figure 2 illustrates a more detailed data flow diagram relating to the schematic representation of Figure 1. Using the database of pre-calculated acceptable fingerprints 16, the system will be able to determine if any given file on a client's system is authentic, i.e. not invaded by rogue software. When comparisons are done, file location, time stamps, platform information, user preferences and other parameters can also be taken into consideration.

The system should be continuously updated with new fingerprint information in the database of acceptable file fingerprints 16 as new software and updates become available. The system thus provides pristine fingerprint information that is made available to the file integrity checking software installed on a client's computer system. Instead of identifying bad files, the system therefore ensures that the data is good. Instead of requiring users to have generated a fingerprint database some time back, the system provides pre-calculated fingerprints and greatly reduces the barriers to adoption of this important file integrity technology.

In addition to the above, the system may also store fingerprints of various customers' files in a separate database (not shown in Figure 1) so that the system can provide heuristic, statistically based best effort guesses on whether a certain fingerprint is acceptable for a given file.

In the above example, in the event of a questionable outcome (categories (b) and (c)) where the system does not have pre-calculated hash information on a

certain file on the client, the system may also render a heuristic result on whether a file is safe. This result can be provided by accessing the second database (not shown in Figure 1) which contains hashes that the customer's administrators have confirmed to be acceptable. For example, if the system does not know about whether a file such as "/usr/bin/myspecialprogram" should have a hash result of "xyz", it can inform the administrator, and also point out one of the following:

1) no other systems in the client's class have such a file so authenticity is uncertain;
2) other systems in the client's class which have such a file do not have a hash of "xyz" so the file is suspicious;
3) other systems in the client's class which have such a file have the same hash of "xyz" so the file is probably safe.

This information can also be provided with a percentage figure so that administrators have a best guess of where they stand before engaging in manual verification as described above.

As the client base grows, this information will allow the system to make increasingly improved guesses. The system can render an opinion along the lines of "10,000 other customers have this file and 9,985 of them have the same fingerprint, so your file is probably safe" - perhaps a common application whose fingerprint that does not already exist in the first database 16. Such information, while not substantive, allows users to zoom into more critical anomalies on their systems sooner. For example, consider this other response: "10,000 other customers have this file and no one has the same fingerprint as you. Worse yet, all these 10,000 customers have the same fingerprint so your file is most probably unsafe." The system can thus provide a percentage or quantifiable risk rating in either a numeric fashion or with the use of colours.

From the client's point of view, the advantage is that even systems currently deployed in risky public network environments can be easily reliably scanned and put onto a file integrity protection regime without re-installation to assure a pristine state and with significantly reduced downtime. As customers apply upgrades to their systems, the system will similarly be able to verify that new

software being installed is authentic since the fingerprints of the new software should be in the system's database 16. Conversely, the system can be programmed to warn the user if the update contains software the system does not believe is authentic.

While a particular embodiment of the invention has been shown and described, it will be obvious to those skilled in the art that changes and modifications of the present invention may be made without departing from the invention in its broader aspects. As such, the scope of the invention should not be limited by the particular embodiment and specific construction described herein but should be defined by the appended claims and equivalents thereof. Accordingly, the aim in the appended claims is to cover all such changes and modifications as fall within the spirit and scope of the invention.

Claims:

1. A method of detecting rogue software including the steps of:

(a) creating a first database containing pre-calculated fingerprints for each file relating to typical operating systems and application software, wherein the pre-calculated fingerprints are calculated using one or more cryptographic formulae;

(b) using the one or more cryptographic formulae to calculate fingerprints of files on a computer system which is to be scanned for rogue software;

(c) comparing the fingerprints calculated for the files on the computer system with the fingerprints which are contained in the first database of pre-calculated fingerprints;

(d) identifying files on the computer system which may contain rogue software by identifying files the calculated fingerprints of which do not correspond to the pre-calculated fingerprints which are stored in the first database.

2. A method according to claim 1 including the further step of generating a list of questionable files on the computer system for which the calculated fingerprints do not correspond to pre-calculated fingerprints stored in the first database.

3. A method according to claim 1, wherein the cryptographic formulae used in the pre-calculation of fingerprints which are stored in the first database and in the calculation of fingerprints for files on the computer system, use one or more hash functions to generate hash values for each file.

4. A method according to claim 1, wherein cryptographic formulae used in the pre-calculation of fingerprints which are stored in the first database and in the calculation of fingerprints for files on the computer system, use one or more asymmetric cryptographic functions to generate digital signatures for each file.

5. A method according to claim 2 wherein questionable files are considered by a system administrator and may be marked by the system administrator as acceptable.

6. A method according to claim 5 wherein the fingerprints for questionable files which are accepted by the system administrator are stored in a second database.

7. A method according to claim 6 which includes the step of calculating the probability that a questionable file has been corrupted by rogue software, by comparing its fingerprint with fingerprints for similar files that have previously been accepted by the system administrator and stored in the second database.

8. A method according to claim 7 wherein the system provides statistical information regarding:

(a) the number of fingerprints in the second database which represent files with the same characteristics as the questionable file;

(b) the number of fingerprints in the second database which are identical to the fingerprint of the questionable file;

(c) the number of fingerprints in the second database which are different to the fingerprint of the questionable file.

9. A method according to claim 5 wherein files that are not acceptable are replaced or reinstalled.

10. A method according to claim 6 wherein the step of calculating fingerprints for files on the computer system and the step of comparing fingerprints on the computer system with corresponding pre-calculated fingerprints stored on the first database are both implemented by the computer system and wherein verification of questionable files takes place before fingerprints from the computer system are added to the second database.

11. A method according to claim 10 wherein the computer system is physically remote from the first database and communication between the

computer system and the first database takes place over a network such as the Internet.

12. A method according to any one of claims 6 to 8 wherein the step of calculating fingerprints for the files on the computer system is implemented by the computer system and the step of comparing the fingerprints which represent files on the client system with corresponding pre-calculated fingerprints stored in the database is implemented by a server, and wherein verification of questionable files takes place between the computer system and the server before the corresponding fingerprints are transferred from the computer system to the second database.

13. A method according to claim 12 wherein the computer system is physically remote from the server and communication between them takes place over a communications network such as the Internet.

14. A system for detecting rogue software including:

(a) a first database containing pre-calculated fingerprints for each file relating to typical operating systems and application software, the fingerprints having been calculated using one or more cryptographic formulae;

(b) a software component which uses one or more cryptographic formulae to calculate fingerprints for files on a computer system; and

(c) a software component which compares the calculated fingerprints for the files on the computer system with corresponding pre-calculated fingerprints stored in the first database, such that files on the computer system which may contain rogue software are identified by identifying files the calculated fingerprints of which do not correspond to the pre-calculated fingerprints which are stored in the first database.

15. A system according to claim 14 including a software component which generates a list of questionable files for which the calculated fingerprints do not correspond to the pre-calculated fingerprints which are stored in the first database.

16. A system according to claim 14 or 15 wherein the software components are installed on the computer system.

17. A system according to claim 14 wherein the pre-calculation of fingerprints which are stored in the first database and calculation of fingerprints for files on the computer system use one or more hash functions to generate hash values for each file.

18. A system according to claim 14 wherein the pre-calculation of fingerprints which are stored in the database and calculation of fingerprints for files on the computer system use one or more asymmetric cryptographic functions to generate digital signatures for each file.

19. A system according to claim 15 further including a second database in which fingerprints of questionable files which are found to be acceptable by a system administrator are stored.

20. A system according to claim 15 wherein the system calculates the probability that a questionable file is a file that has been corrupted by rogue software, by comparing its fingerprint with fingerprints for similar files that have been verified and stored in the second database.

21. A system according to claim 20 wherein the system produces statistical information regarding:

(a) the number of fingerprints in the second database which represent files with the same characteristics as the questionable file;

(b) the number of fingerprints in the second database which are identical to the fingerprint of the questionable file;

(c) the number of fingerprints in the second database which are different to the fingerprint of the questionable file.

22. A system according to claim 19 wherein files that are not acceptable are replaced or reinstalled.

23. A system according to any one of claims 14 to 22 wherein the step of calculating fingerprints of files on the computer system and the step of comparing the fingerprints on the computer system with corresponding pre-calculated fingerprints stored on the database are both implemented by the computer system and wherein verification of questionable files takes place before fingerprints from the computer system are added to the second database.

24. A system according to claim 23 wherein the computer system is physically remote from the first database and communication between the computer system and the first database takes place over a network such as the Internet.

25. A system according to any one of claims 14 to 20 wherein the step of calculating fingerprints for the files on the computer system is implemented by the computer system and the step of comparing the fingerprints which represent files on the computer system with corresponding pre-calculated fingerprints stored in the first database is implemented by a server and wherein verification of questionable files takes place between the computer system and the server before the corresponding fingerprints are transferred between them.

26. A system according to claim 25 wherein the computer system is physically remote from the server and communication between them takes place over a communications network such as the Internet.

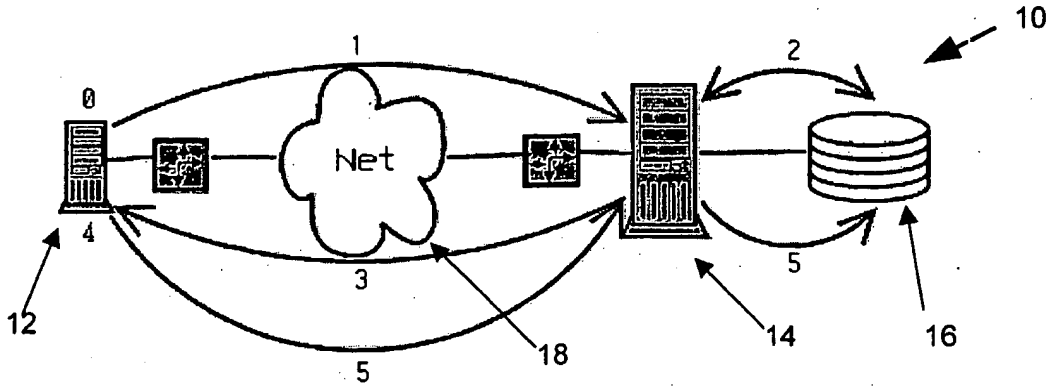


FIGURE 1

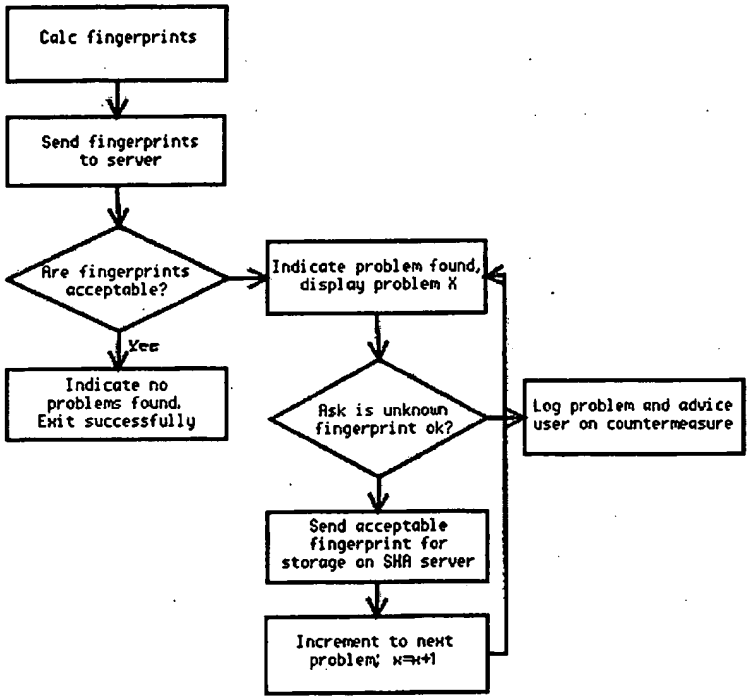


FIGURE 2

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
25 April 2002 (25.04.2002)

PCT

(10) International Publication Number
WO 02/033525 A3

(51) International Patent Classification[7]: G06F 1/00

(21) International Application Number: PCT/SG01/00213

(22) International Filing Date: 17 October 2001 (17.10.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
200005973-3 17 October 2000 (17.10.2000) SG

(71) Applicant and
(72) Inventor: CHUANG, Shyne-Song [SG/SG]; Singapore Post Centre, P.O. Box 493, Singapore 914017 (SG).

(74) Agent: DREW & NAPIER LLC; 20 Raffles Place, #17-00 Ocean Towers, Singapore 048620 (SG).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

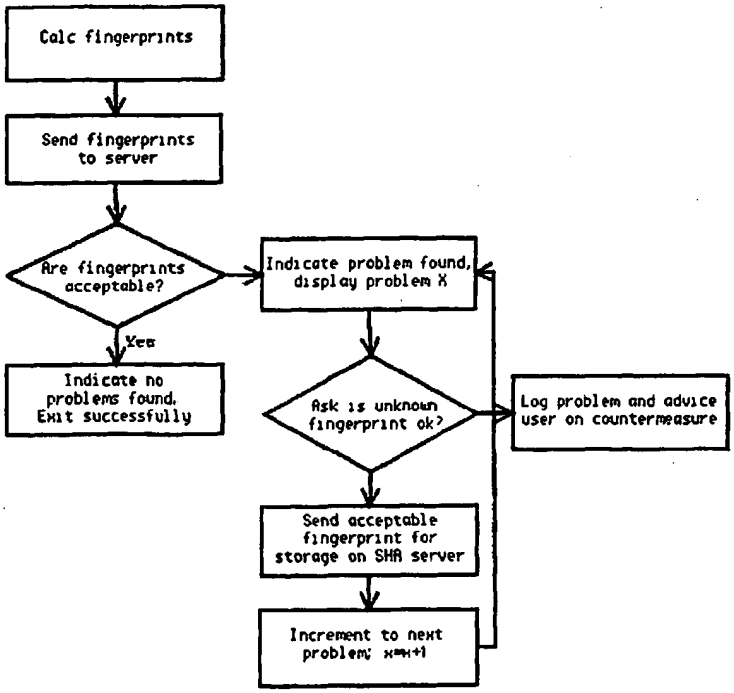
(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

(88) Date of publication of the international search report:
6 March 2003

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: A METHOD AND SYSTEM FOR DETECTING ROGUE SOFTWARE



(57) Abstract: A method of detecting rogue software includes the step of creating a first database containing pre-calculated fingerprints for each file relating to typical operating systems and application software, wherein the pre-calculated fingerprints are calculated using one or more cryptographic formulae. The one or more cryptographic formulae are then used to calculate fingerprints of files on a computer system which is to be scanned for rogue software. The fingerprints calculated for the files on the computer system are compared with the fingerprints which are contained in the first database of pre-calculated fingerprints. Files on the computer system which may contain rogue software are identified by identifying files the calculated fingerprints of which do not correspond to the pre-calculated fingerprints which are stored in the first database.

WO 02/033525 A3

Trend Micro, Inc.
EX1007-626

INTERNATIONAL SEARCH REPORT

Int al Application No
PCT/SG 01/00213

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F1/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WILLIAMS R N: "DATA INTEGRITY WITH VERACITY" INTERNET, 'Online! 12 September 1994 (1994-09-12), XP002096828 Retrieved from the Internet: <URL:ftp://ftp.rocksoft.com/clients/rocksoft/papers/vercty10.ps > 'retrieved on 1994-09-12!	1-6,9, 14-20,22
Y	page 4, left-hand column -right-hand column, paragraph 3 page 5, left-hand column, paragraph 6 - paragraph 9 page 6, right-hand column, paragraph 6 -page 8	7,8, 10-13, 20,21, 23-26

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
29 November 2002	09/12/2002

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Powell, D

Form PCT/ISA/210 (second sheet) (July 1992)

Trend Micro, Inc.
EX1007-627

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
	page 10	
X	"MECHANISM FOR TRUSTED COMPUTING BASE DEFINITION AND CHECKING" IBM TECHNICAL DISCLOSURE BULLETIN, IBM CORP. NEW YORK, US, vol. 34, no. 9, 1 February 1992 (1992-02-01), pages 188-191, XP000300643 ISSN: 0018-8689 page 188, paragraph 3 - paragraph 4 page 191	1-5,9, 14-19,22
Y	US 6 021 491 A (RENAUD BENJAMIN J) 1 February 2000 (2000-02-01) abstract column 3 -column 4; figures 1,3,4 column 5, line 7-23 column 6, line 29-64 column 7 -column 8	7,8, 10-13, 20,21, 23-26
A	WO 00 28420 A (SYMANTEC CORP) 18 May 2000 (2000-05-18)	

Trend Micro, Inc.
EX1007-628

INTERNATIONAL SEARCH REPORT

Int ial Application No
PCT/SG 01/00213

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 6021491 A	01-02-2000	EP 0845733 A2	03-06-1998
		JP 10326078 A	08-12-1998
		US 5958051 A	28-09-1999
WO 0028420 A	18-05-2000	US 6094731 A	25-07-2000
		EP 1129406 A1	05-09-2001
		WO 0028420 A1	18-05-2000

Form PCT/ISA/210 (patent family annex) (July 1992)

OIPE TAPS
JAN 12 2007
PATENT & TRADEMARK OFFICE

Docket Number: 011765-0355986 PATENT APPLICATION
Client Reference: AF/JG/P10717US

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re the Application of

MELVYN MORRIS, et al. Group Art Unit: 2131

Application No.: 11/477,807 Examiner: Unassigned

Filed: June 30, 2006 Confirmation No.: 4785

For: METHODS AND APPARATUS FOR DEALING WITH MALWARE

INFORMATION DISCLOSURE STATEMENT

Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

Sir:

Pursuant to 37 CFR 1.56, the attention of the Patent and Trademark Office is hereby directed to the references listed on the attached PTO-1449. Copies of the required references are provided herewith. It is respectfully requested that the information be expressly considered during the prosecution of this application, and that the references be made of record therein and appear among the "References Cited" on any patent to issue therefrom. Applicants respectfully request the Examiner return an initialed copy of the enclosed Form PTO-1449 to Applicants with the next Office communication to indicate that the references have been considered, per MPEP § 609.

This Information Disclosure Statement is being within three months of the mailing date of the International Search Report in the related international application. No fee is required.

Respectfully Submitted,
PILLSBURY WINTHROP SHAW PITTMAN LLP

Christine H. McCarthy
Registration Number 41844
Customer Number: 00909

Date: January 12, 2007
Telephone: (703) 770-7900
Facsimile: (703) 770-7901
P.O. Box 10500
McLean, VA 22102

Trend Micro, Inc.
EX1007-630

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 315 066 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
28.05.2003 Bulletin 2003/22

(51) Int Cl.[7]: **G06F 1/00**, H04L 29/06

(21) Application number: **01309795.1**

(22) Date of filing: **21.11.2001**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR
Designated Extension States:
AL LT LV MK RO SI

(71) Applicant: **BRITISH TELECOMMUNICATIONS public limited company**
London EC1A 7AJ (GB)

(72) Inventor: **The designation of the inventor has not yet been filed**

(74) Representative: **Nash, Roger William et al**
BT Group Legal Services,
Intellectual Property Department,
Holborn Centre, 8th Floor,
120 Holborn
London EC1N 2TE (GB)

(54) Computer security system

(57) A computer security system for use in a network environment comprising at least a plurality of user computers arranged to communicate over a network, the system comprising a warning message exchange system operable to allow the communication from the user computers of warning messages relating to suspect data identified as a possible security threat; a message counting system operable to maintain a count for every particular piece or set of suspect data based on the number of warning messages communicated relating thereto; and network security means operable to act against any particular piece or set of suspect data for which the count maintained therefor exceeds at least one threshold value

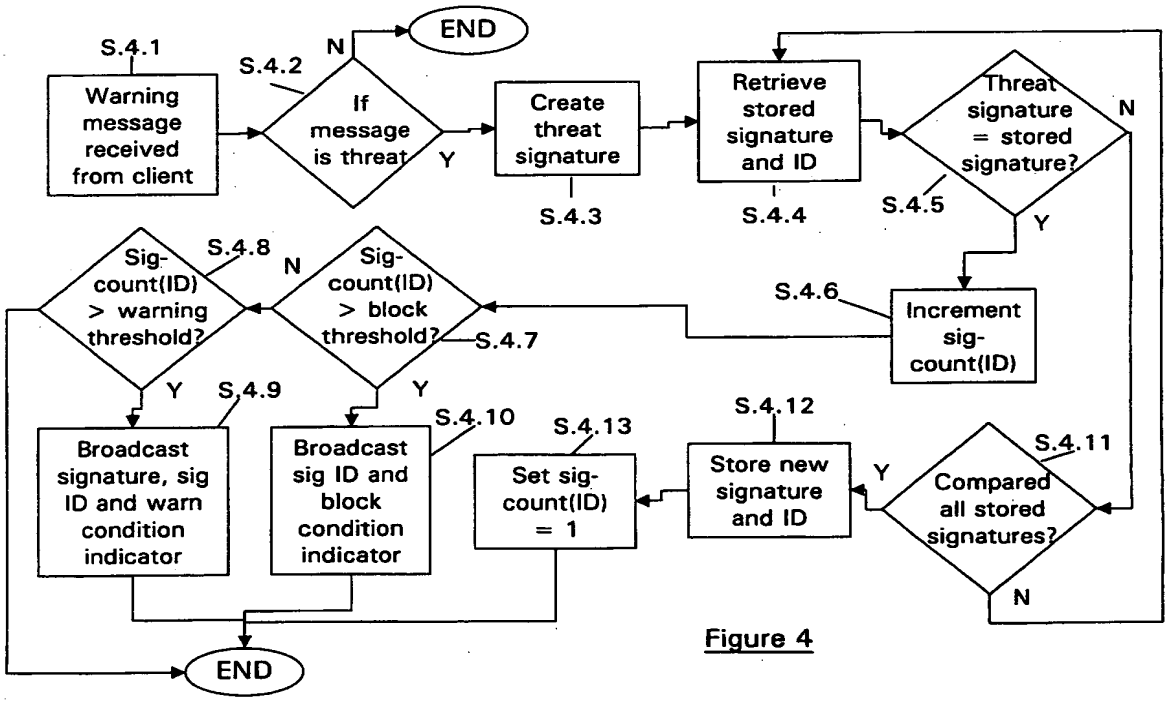


Figure 4

Printed by Jouve, 75001 PARIS (FR)

EP 1 315 066 A1

Trend Micro, Inc.
EX1007-631

Description

Technical field

[0001] The present invention relates to a computer security system for use in a networked environment, the system providing a measure of security against suspect data such as computer viruses and the like.

Background to the Invention

[0002] Malicious attacks on computer systems have plagued individuals and organisations for many years. Initially these attacks mostly took the form of viruses in which a rogue program infected a machine via an infected floppy disk or network location for example. More recently so called worms and Trojan horses have caused much disruption in which the user is deceived into running a program via an attachment to an e-mail or a rogue program masquerading as something more innocuous. A large industry has grown up around protection from such attacks. These companies provide anti-virus software that typically resides on individual machines and monitors the system to check for the presence of known viruses. These viruses can take many forms - some more advanced forms are fully polymorphic in that the byte code "signature" is entirely different from one instance of the virus to the next. This is achieved through the use of encryption technology and/or the addition of spurious and random code to the virus. However, the majority of viruses that are in the "wild" and the cause of costly disruption to computer systems are relatively simple and can be detected by simple byte-code signature matching. Many of the current anti-virus programs use just such techniques and are successful if the signature of a discovered virus can be delivered to machines before the virus strikes.

[0003] The operation of such anti-virus programs and systems is well known in the art, and is usually as follows. A computer user will have installed on their computer an anti-virus program which is provided with a database of known computer virus byte-code signatures. The program will tend to run in the background continuously monitoring operations performed on the computer, and data received at and transmitted from the computer. If the byte-code signature of a known virus stored in the anti-virus program's database is detected by the program, the anti-virus program informs the user and takes appropriate action against the data, such as deleting it or storing it in a protected drive.

[0004] Such prior approaches tend to suffer from a big problem, however - delay. Even the simplest of viruses and worms may still cause great disruption at enormous cost to organisations. This is because the process from discovery of a new virus to delivering its signature to all protected machines takes too long, and requires an administrative authority such as the anti-virus program manufacturer (or in some cases an organisation's IT department) to recognise the problem, take action to identify the virus's signature, update the anti-virus database, and distribute the updated database to each user. By the time such a sequence of actions is complete the damage has already been done in many cases. What is required is a much more rapid approach - one that can operate on the same time scales as the spread of the virus and thus provide much more rapid and cost-saving protection.

Prior Art

[0005] There is much work exploring the use of agent technology to provide rapid reaction to malicious attack. In this work computer systems are inhabited by a number of agents whose job it is to detect intrusions. One interesting approach aims to draw metaphor from immune systems, which allow an organism to rapidly respond to previously unseen foreign objects through a "real-time" evolutionary system, as disclosed in Artificial Life IV, Proceedings of the Fourth International Workshop on Synthesis and Simulation of Living Systems, Rodney A. Brooks and Pattie Maes, eds., MIT Press, Cambridge, Massachusetts, 1994, pp. 130-139 These kinds of agent systems may prove useful in future security systems, but rely on the successful development and deployment of software agents which can detect intrusions. Many other potential future anti-virus systems are also discussed at http://www.research.ibm.com/antivirus/SciPapers.htm. In the meantime until the potential of these ideas is exploited, there is still a need for a system which allows for more rapid prevention of the spread of computer viruses.

[0006] One such prior art system which claims to increase response times is that of the Symantec® Digital Immune System (DIS) , which is described at http://securityresponse.symantec.com/avcenter/reference/dis.tech.brief.pdf. The DIS operates by providing an heuristic search engine which automatically identifies possible viruses at either the user's desktop, the server, or the network gateway. A secure channel is then provided direct to Symantec's security response centre over which the suspect data can be transmitted. At the security response centre the received data is subject to a number of automatic filters to detect, for example, known clean files, files which are known to return false positives from the heuristic search engine, and known virus files. If the suspect data is not filtered out by any of these filters then it is directed to an analysis program for further analysis and action.

[0007] The DIS system therefore allows for the automatic identification, secure submission, automatic prioritisation, and subsequent analysis of suspect data, but ultimately still involves a central authority in the loop to analyse suspect files which make it through the filters. Thus, whilst DIS may improve response times to virus infection through it's automatic filtering processes, it still relies on a central authority to analyse the suspect data

Trend Micro, Inc.
EX1007-632

and decide on appropriate action, which must then be communicated outwards to each user. There is therefore still a need for a system which removes this centralised analysis step to speed the response.

Summary of the Invention

[0008] In order to address the above problems, the present invention provides a collaborative computer security system wherein the responsibility for detection of malicious data such as a computer virus is removed from that of any central authority, and is instead placed in the hands of each and every user of the network. More particularly, the present invention provides a distributed virus identification system which allows individual users or software agents running on a user's computer to identify malicious data when they receive it, and to pass a warning message concerning the received data to either their peers, or to a central server. A record is kept either at the server or at each peer computer as to the number of warning messages communicated concerning any particular piece or set of suspect data, and then appropriate security actions such as issuing warnings to users or blocking the transmission of the suspect data can be taken once the number of warning messages communicated from users has passed a certain threshold level. The advantages of the invention are that an organisation's response to a computer virus intrusion can be much more rapid than was previously the case where a central authority had to identify the problem and take action.

[0009] In view of the above, the present invention provides a computer security system for use in a network environment comprising at least a first group of user computers arranged to communicate over a network, the system comprising a warning message exchange system operable to allow the communication from the user computers of warning messages relating to suspect data identified as a possible security threat; a message counting system operable to maintain a count for every particular piece or set of suspect data based on the number of warning messages communicated relating thereto; and network security means operable to act against any particular piece or set of suspect data for which the count maintained therefor exceeds at least one threshold value

[0010] The primary advantage of the invention is that it allows the security system to work on the same time scales as the spread of the attack. The more the computer virus or worm spreads the more likely it is that enough positive identifications are made to allow its signature to be ascertained and protection to be installed. This overcomes the problem of traditional security systems where an authority has to learn of the virus, take action, and then distribute associated protective measures to the users.

[0011] A further advantage of the system is that it allows a lot of information to be rapidly collected about a computer virus. For example, current "worms" that are "in the wild" are mostly identical at the byte code level, however it is likely that in the future this will no longer be the case, and that there will be variations in different instances of the same worm. In such a case, the more instances of the worm that can be collected the more likely that it will be that suitable protection can be devised. By allowing for individual identification of a worm at each computer by the user, the present invention allows for multiple instances of the same virus to be identified and collected.

[0012] The one or more "threshold values" against which the respective counts for pieces or sets of suspect data are compared may be any value, and may be fixed or dynamically adaptive. Furthermore the actual values may be generated by any appropriate function, such as for example by a random number generator, or as a function of some metric of accuracy of users in identifying viruses. Furthermore, in other embodiments it is foreseen that the threshold values may not simply be actual numbers which the respective counts must be greater than, but could instead be probabilistic in nature. Furthermore, various logical functions or rules can be used for their creation, including for example, fuzzy logic.

[0013] Preferably, the judgement as to whether any piece or set of data present at any one or more of the user computers is a possible threat is performed by the respective human user or users of the or each user computer. Such an arrangement exploits the massive untapped resources and expertise of many computer users, and again allows for a rapid response to the spread of a computer virus.

[0014] Preferably, the network security means is further operable to compare the maintained count for a particular piece or set of suspect data against a plurality of different thresholds, and to take different action against the data dependent upon which threshold or thresholds are exceeded.

[0015] Such an approach allows the computer security systems to have a graded response to a computer virus intrusion, and to take different actions depending on the number of warnings received from users. Furthermore, it is also possible to weight warnings from different users, such that a warning from a user who is particularly expert in the field is weighted more heavily than a casual user of the system.

[0016] Preferably, the action taken by the network security means comprises warning each of the users as to the suspect nature of a particular piece or set of data. Furthermore, the action may also comprise preventing the transmission of the particular piece or set of suspect data across the network. Usually, the warning action will be taken when the number of messages communicated exceeds a lower threshold than a second threshold which must be crossed in order for the transmission blocking action to take place.

[0017] Preferably, the or each warning message com-

Trend Micro, Inc.
EX1007-633

prises at least an identifier of the piece or set of suspect data to which it relates. The identifier may be the actual piece or set of suspect data itself, or a repeatably generatable signature substantially unique to the piece or set of data. By allowing for the collection of such identifiers, the security system also provides a means of collecting information on multiple computer viruses, and also on adaptive computer viruses whose byte code signature may change in each instantiation on a users machine.

[0018] In a preferred embodiment, the computer security system further comprises a network server arranged to receive the or each warning message communicated from the user computers, and further arranged to host the message counting system and at least part of the network security system, wherein that part of the network security system hosted by the server is operable to determine the action which should be taken against a particular piece or set of suspect data, and to communicate an action condition indicator indicative of the determined action to each of the plurality of users.

[0019] By using a central server to maintain the warning message counts and to receive the warning messages themselves, the network traffic generated by the computer security system is minimised, as users need only send a warning message direct to the server, which can then broadcast warnings about suspect data as appropriate to all the users.

[0020] In another embodiment, the warning messages generated by a user computer are broadcast to every other user computer, and each user computer is further arranged to host its own message counting system and network security system operable in accordance with any of the above. Such an approach represents a "peer-to-peer" system, and provides advantages that it is not dependent upon a central server for its operation, thereby presenting a more robust distributed system. Such a peer-to-peer system has a potential disadvantage, however, in that depending on the exact implementation it may be that the network traffic required to broadcast the warning messages from each user to every other user is unacceptably high.

[0021] To at least partially overcome this problem, in another embodiment there is provided an inter-group communications system operable to allow the communication of warning messages relating to suspect data identified as a possible security threat from the first group of user computers to one or more other groups of other user computers. By splitting the user computers intro groups, the number of warning messages which are required to be transmit is substantially reduced, thereby conserving network signalling bandwidth.

[0022] Preferably, the inter sub-set warning messages are transmitted by the inter-sub-set communications system if the number of warning messages communicated by the user computers in the sub-set concerning a particular piece or set of data exceeds at least one threshold value. Preferably the threshold value or values is/are the same as those at which the network security system takes action.

[0023] In further embodiments, the message counting system is further arranged to store one or more weighting coefficients relating to one or more particular user computers, and to increment the count maintained for a particular piece or set of suspect data by an amount based upon the weighting coefficient when a warning is received from the one or more particular user computers relating to the particular piece or set of suspect data

[0024] From a second aspect, the present invention also provides a computer readable storage medium storing one or more computer programs which when run on a computer cause the computer to operate in the manner of the computer security system as described above. Such a computer security system may provide all the additional features and advantages mentioned previously.

Brief Description of the Drawings

[0025] An overview of the operation of the invention, together with a description of a number of embodiments thereof, presented by way of example only, will now be made with reference to the accompanying drawings, wherein like reference numerals refer to like parts, and wherein:-

Figure 1 illustrates a typical network environment in which the computer security system of the present invention is intended to operate;

Figure 2 illustrates a computer storage medium located within each users computer, and depicts the programs and data stored thereon in a first embodiment of the invention;

Figure 3 depicts a computer storage medium located at a server used within the first embodiment of the inventions, and further depicts the programs and data stored thereon;

Figure 4 is a flow diagram showing the operation steps performed by the warning creation program stored at the server in the first embodiment of the present invention;

Figure 5 is a flow diagram showing the method steps performed by the warning receipt program stored at a users computer in the first embodiment;

Figure 6 is a flow diagram showing the steps performed by a filter program in either of the embodiments of the invention;

Figure 7 depicts a computer readable storage medium located at a users computer in a second embodiment of the present invention, and further depicts the programs and data stored thereon;

Figure 8 is a flow diagram of the steps performed by a warning creation program in the second embodiment of the present invention;

Figure 9 is a flow diagram of the steps performed by a warning receipt program in the second embod-

Trend Micro, Inc.
EX1007-634

iment of the present invention;
Figure 10 is a flow diagram illustrating the steps performed by one of the functions in both Figures 8 and 9; and
Figure 11 illustrates an organisational architecture of user computers which may be used in another embodiment of the invention.

Overview of the operation of the invention

[0026] The present invention recognises and makes use of the fact that a computer system is already inhabited by a number of intelligent agents - the users themselves. In order to summarise the operation of the invention consider an attack by an e-mail "worm" (a type of computer virus such as the "Melissa" virus which was encountered in 2001). A worm is sent to an employee of a given organisation. With some probability the employee opens the e-mail attachment and unwittingly unleashes the worm that causes some damage to the individuals machine and propagates itself to a number of other machines through use of the individuals e-mail system and address book. In each of these cases there is some probability that the user will open the attachment and further propagate the virus. Thus, the statistics are in the worm's favour - not everyone will open the attachment but it only takes a small fraction to do so for the worm to propagate throughout an organisation's computer system. However, for every user who opens the attachment there are likely to be many more who correctly recognise the e-mail as a potential threat and discard it. Vital information is thus gained in this process - an intelligent agent has successfully detected an intrusion - and it is this information that the present invention exploits to its full effect. Doing so opens up the potential for the statistics to be turned against the worm and in favour of the protection system.

[0027] Within the invention the user informs the system of the recognition by forwarding the mail or passing the mail or other program to an anti-virus program that is running on that machine, which then passes it to the server (or in an alternative embodiment direct to the user's peers). The signature of the worm is then quickly ascertained and a count kept of the number of warnings received from users about the worm. Once the count reaches a certain threshold the signature of the worm is distributed to all the other individual machines, who then use the signature to filter any received data or mail. As an alternative the server itself could simply filter out all e-mails with such a signature. In either case the technique of allowing users (or user machines provided with appropriate software agents) to detect and provide warning of the worm aligns well with the trends in the industry towards peer to peer systems where the vast untapped resources at the edges of the network are beginning to be used to greatly enhance storage and computational power for example. The invention is therefore essentially a peer to peer security system that may tap the vast unused resource of the user's knowledge and experience, or the user computer's processing power.

Description of the embodiments

[0028] Two specific embodiments of the invention will now be described in detail, followed by description of various alterations that may be made to each to provide additional embodiments. Both main embodiments of the invention implement a computer security system for use in a network environment. A typical network environment in which the invention may be used is depicted in Figure 1.

[0029] Figure 1 illustrates a typical local area network (LAN) configuration. Here, a plurality of user computers 15 are connected to a network 10 which is another control of a network server 12. The network 10 provides typical network functionality, such as the ability to allow users 15 to communicate with each other and with the server over the network. The network 10 in the invention may be any particular network technology, may be wired or wireless, and may be scaled to any network size or architecture known in the art. Therefore, whilst the present invention is particularly useful for use within individual organisations provided with a LAN, it may also be used on a wider scale in a metropolitan area network (MAN) or a wide area network (WAN). Within the embodiments, each of the user computers 15, the network itself 10, and the server 12 are provided with appropriate equipment and software to permit the usual network communication functions as are known in the art.

[0030] Although the above makes reference to the user computers 15 being part of the same network, it should be noted that it is not essential that each user-computer be located in the same actual network (although this may be preferable). This is because all that the invention requires is the ability for any particular user computer to be able to send messages either to its peers or to a server, and hence all that is required is some message transport mechanism between such parties. Such a message transport mechanism may be a single network to which all the user computers and servers (where required) are connected, or may be a sequence of interconnected networks such as the Internet, with different user-computers actually being part of different networks.

[0031] Having described the context in which the invention is intended to operate, a first, preferred, embodiment which represents the best mode of the invention will now be described with reference to Figures 2 to 6.

[0032] The first embodiment of the invention provides a computer security system wherein identification of suspect data is performed by the users or suitable software agents installed and running on the user's computers at the user's computers themselves, and upon identification of a suspect piece or set of data a warning message is transmitted from the user's computer to the network server 12. At the server the number of warning

Trend Micro, Inc.
EX1007-635

messages received about a particular piece or set of data is counted, and once the count passes a first warning threshold, a warning message is broadcast to all users, the message containing a signature of the suspect data, such that an anti-virus program located at each user's computer can filter incoming data for the suspect data. If further warnings are received from users by the server, and the count of warning messages exceeds a second threshold, then the server broadcasts a second message to all of the users on the network instructing the anti-virus programs located at each user computer to block the suspect data to prevent any onward transmission or infection.

[0033] Given the above overview, within the first embodiment each user computer 15 contains a computer readable storage medium 22, as shown in Figure 2. The computer readable storage medium will typically be a magnetic hard drive, or may equally be a DVD RAM, DVD-RW, DVD+RW, solid state memory, optical disc, magneto optical disc, or the like. The computer readable storage medium 22 at each user computer 15 is arranged to store a data filter program 24, and a warning receipt program 26. In addition, storage space is also provided for suspect data signatures and ID's, stored in the form of a data base 28.

[0034] The server 12 within the first embodiment also contains a computer readable storage medium 32 which will typically be a magnetic hard disc, but may also be any of the storage devices mentioned previously in respect of the or each user computer 15. Within the first embodiment the computer readable storage medium 32 stores a warning creation program 34, and a suspect data signature and ID database 36. Optionally, the computer readable storage medium 32 may also store a data filter program 38, to provide an alternative method of operation as will be described later.

[0035] Having described the main system hardware elements required by the first embodiment, the operation thereof will now be discussed with reference to the flow diagrams of Figures 4, 5, and 6.

[0036] Assume that a user computer 15 connected to the network 10 receives a message over the network which contains a computer virus or the like (referred to hereafter as suspect data). Within the first embodiment of the invention a user of the user computer 15 upon reading the message recognises that the message contains such suspect data, and hence within the first embodiment then forwards the suspect data as the payload in a warning message of predetermined format to the server 12. By forwarding the data as the payload encapsulated in a warning message of known format, the server knows that the payload is potentially suspect data, and hence will not become victim to the computer virus itself. At the server, upon receipt of such a warning message from one of the user computers 15, the warning creation program 34 is run, and the steps performed by this program are shown in Figure 4.

[0037] Firstly, at step 4.1 the server receives a warning message from one of its clients, being one of the user computers 15 (as described above). Then, at step 4.2 the server parses the warning message to retrieve the suspect data, and checks the message to see if it could be a computer virus. Such checking preferably takes the form of determining the data type to see if it could feasibly be a virus. Viruses are frequently executable files, such as .exe or .bat on a PC, or macros such as are found in Microsoft® Word® or Excel®. Such a check can conveniently be performed by looking at any file extension on the data. If such checking does not indicate that the data could be a virus then processing ends, as it is assumed that the user has simply made a mistake. Instead, if the warning message is in fact a true warning and the suspect data could possibly be a virus, then processing proceeds to step 4.3.

[0038] Note here that the checking step of step 4.2 is performed in the preferred embodiment, but in other embodiments may not be performed, and as such is optional. Where step 4.2 is not performed processing proceeds directly from step 4.1 to 4.3.

[0039] At step 4.3 the server reads the suspect data from the payload, and creates a "signature" for the suspect data, to permit the particular piece or set of suspect data to be identified in the future. Such a signature may be conveniently created by running the suspect data through a hash function. Hash functions are well known in the art, and commonly provide a 16 or 20 bit output word unique to any particular piece or set of data input thereto. The key feature of hash functions which may be of use in the present invention, however, is that the same output word is always output from the hash function for the same piece or set of input data. Thus the hash value for a certain piece or set of suspect data will always be the same, no matter where the suspect data has been received from.

[0040] It should be noted that other forms of signature creation other than the use of hash functions may also be of use, the only requirement being that an identifiable unique signature is generated for any particular piece or set of suspect data input into the signature creation function at any time.

[0041] Following the creation of the data signature, at step 4.4 the signature and ID database 36 is accessed to retrieve the first previously stored signature and accompanying signature ID therefrom. The signature and ID database 36 in the server 12 is used to store the generated signature ID's from pieces of suspect data received in previous warning messages. At step 4.4 the first signature and ID in the data base is retrieved the first time the controlling function for step 4.4 is called, and thereafter at succeeding calls the next signature and ID in the database list are successfully retrieved.

[0042] At step 4.5, the threat signature of the received suspect data generated at step 4.3 is compared with the stored signature retrieved from the database at step 4.4. If the threat signature is not the same as the retrieved stored signature then processing proceeds to step 4.11,

Trend Micro, Inc.
EX1007-636

wherein a check of the signature and ID database 36 is made to see if all stored signatures have been compared with the new signature. If not then processing proceeds to step 4.4, which retrieves the next signature and ID from the database 36 for evaluation at step 4.5 with the generated data signature for the presently received.

[0043] If at step 4.11 it is determined that all of the stored signatures have been compared with the newly created threat signature then it must be the case that the warning message received from the user computer at step 4.1 is the first warning message from any of the user computers to bring to the server's attention the particular piece or set of suspect data contained in the warning message payload. In this case it is necessary for the server to store information on the suspect data, and this is achieved by storing the suspect data signature created at step 4.3 together with a new unique signature ID within the signature and ID database 76. This storing function is accomplished at step 4.12, and thereafter processing proceeds to step 4.13, wherein a count of warning messages received for the particular piece of suspect data is established, and initiated to a value of one. It will be appreciated that a separate count is maintained for every different piece or set of suspect data received in a warning message, and hence at any one time the server will be maintained in several different counts. Therefore, conveniently the various counts may be maintained in a one dimensional array structure, indexed by the particular suspect data's signature ID. Such storage of the count is shown in step 4.13.

[0044] Following the storage of the data signature, signature ID, and establishment of the message count for the suspect data, then in the case that the suspect data had never been received before processing then ends.

[0045] Returning to a consideration of step 4.5, if it turns out that the particular piece or set of suspect data included in the warning message has been received before in a previous warning message then at some point in the loop formed by steps 4.4, 4.5, and 4.11, the evaluation performed at step 4.5 will return a positive value, in that the threat signature generated at step 4.3 will match a previously stored signature. In this case processing then proceeds to step 4.6, wherein the ID of the stored signature is used at an index into the signature count array, and the particular count for the stored signature is incremented by one to reflect the receipt of an additional warning message relating to the particular piece or set of suspect data to which the signature relates. Following the incrementing of the signature count, processing proceeds to step 4.7.

[0046] Step 4.7 represents the first thresholding evaluation performed to determine whether or not a sufficient number of warnings have been received from the user computers concerning a particular piece or set of data such that action should then be taken against that data. Therefore, at step 4.7 an evaluation is performed to determine whether the signature count for the particular piece or set of suspect data is greater than a "block" threshold value. If so processing proceeds to step 4.10, wherein the server then broadcasts a message to all of the user computers 15 on the network, the message containing the particular data signature ID, and a message condition indicator field which in this case contains data instructing the user computers to block the suspect data if it is received thereat. Blocking of the suspect data is performed by the data filter program 24 provided at each user computer 15, the operation of which is described later.

[0047] It should be noted that at step 4.10 the message from the server to the clients contained only the signature ID and the condition indicator and does not contain the actual signature itself. The reason for this is that it is assumed that prior to any particular piece or set of suspect data becoming blocked a warning message will have been transmitted from the server to the user computers containing the suspect data signature, and hence there is no need to retransmit the signature at step 4.10.

[0048] If the evaluation at step 4.7 returns a negative, processing proceeds to the evaluation at step 4.8 wherein the signature count for the particular piece or set of suspect data is compared against a warning threshold level. It should be noted here that the warning threshold level is set less than the block threshold level such that the evaluation of step 4.8 is always found positive before the evaluation of step 4.7. If the evaluation of step 4.8 returns a negative such that the count is less than the warning threshold then this means that not enough warning messages have been received from user computers concerning the particular piece or set of suspect data such that no action should be taken as yet. In this case processing of the warning creation program 34 then ends.

[0049] In contrast, if the evaluation at step 4.8 is positive, the processing proceeds to step 4.9 wherein the server broadcasts a message over the network to all the user computers 15, the message containing the suspect data's signature as generated at step 4.3, a new data signature ID, being simply an identifier of the signature which can be used as a short hand means of identifying the particular piece or set of suspect data, as well as a "warn" flag in a data condition field of the message. The presence of the warn flag in the data condition field of the message causes the data filter program 24 at the user computer to warn the user whenever the suspect piece of data is received, as will be described later. Following the broadcast of the warning message at step 4.9, processing by the warning creation program 34 ends.

[0050] The output of the warning creation program 34 is therefore a warning message broadcast to all the user computers 15 instructing the user computers to upgrade their threat levels for a particular piece or set of suspect data provided that enough warning messages have been received at the server concerning the particular

Trend Micro, Inc.
EX1007-637

suspect data. At each user computer upon receipt of a warning message from the server the warning receipt program 26 is run, and the steps performed thereby are shown in Figure 5.

[0051] With reference to Figure 5, at step 5.1 a warning message containing the information described earlier is received from the server. Following this, at step 5.2 the user computer 15 passes the warning message to determine whether or not the message contains a data signature.

[0052] If the message does not contain a data signature, then this means that a warning message must have been received at some point in the past from the server containing the data signature, and therefore the data signature will be already stored in the ID database 28 at the user computer 15. In this case processing proceeds to step 5.4, wherein the signature ID contained within the received warning message is used to index and retrieve the appropriate signature from the database 28, whereupon processing then proceeds to step 5.6.

[0053] At step 5.6 the data condition indicator which is contained in the warning message received from the receiver is read, and a condition field in the database 28 related to the particular suspect data signature is updated to match the condition indicated in the warning message from the server. In this case, as a warning message will have already been received concerning the particular suspect data and hence a "warn" condition already set, it is likely that the condition is being updated to a "block" condition. Following step 5.6 the processing by the warning receipt program ends.

[0054] If at step 5.2 it is determined that the warning message from the server does contain a data signature, then this will be the first warning message received from the server concerning the particular suspect data. In this case, processing proceeds to step 5.3 wherein the received signature is stored in the database 28 indexed by the ID which is also contained in the received message. Furthermore, a condition field in the database related to the signature is also set, at step 5.4. In this case, as this will be the first warning message received from the server about the particular suspect data the condition will be set to a "warn" condition. Following step 5.4 the processing by the warning receipt program 26 ends.

[0055] It has thus been described how users can generate warnings about suspect data which are then transmitted to the server, which then broadcasts warnings to all users if enough user's specific warnings have been received. Users can then record the particular security advice broadcast from the server concerning a piece or set of suspect data, and we now describe how user computers 15 may use the broadcast warnings within the data filter program 24.

[0056] More particular, as mentioned previously each user computer 15 is provided with a data filter program 24 which is arranged to run in the background of the computer all the time the computer is operating in a similar manner to anti-virus and other security programs of the prior art. The data filter program 24 acts to monitor all data received at the user computer 15 upon which it is stored to determine if any of the received data matches the signatures of any suspect data stored in the signature and ID database 28. The steps performed by the data filter program 24 in performing this function are shown in Figure 6.

[0057] At step 6.1 assume that the user computer 15 has received some data in the form of an email message or the like over the network 10. At step 6.2 the data filter program 24 acts to create a signature of the received data using the same signature generation function as is used in the server 12. That is, the signature of the received data may be created using a hash function or indeed any other signature generating function which provides a repeatably generatable unique signature for a given set of input data.

[0058] Following the creation of the data signature, at step 6.3 the signature and ID database 28 at the user computer 15 is accessed and the first signature and signature ID within the database is retrieved therefrom. Upon subsequent calls to the controlling function of step 6.3 the next signature and signature ID in the database list is retrieved successively.

[0059] Following step 6.3 processing proceeds to the evaluation of step 6.4, wherein the data signature created at step 6.2 is compared with the retrieved signature from the database 28. If these signatures do not match then processing proceeds to the evaluation of step 6.10, wherein the database is again accessed to see if all signatures have been accessed and compared. If at step 6.10 it turns out that all of the signatures stored in the database have been compared against the signature of the received data and no match has been found, then it can be concluded that the received data is not suspect data, and hence the data filter program ends. In such a case the user is then free to use the received data in whatever manner she chooses.

[0060] In contrast, if at step 6.10 it is determined that not all of the stored signatures have been compared against the received data signature, processing returns to step 6.3 wherein the next stored signature and ID in the database are retrieved, and the evaluation of 6.4 is performed once again.

[0061] If at step 6.4 it is determined that the received data signature is the equivalent of a previously stored signature then processing proceeds to step 6.5. Here, the database is again accessed to retrieve the signature condition associated with the stored signature to determine what security condition was attached to the signature by the server in the last warning message received from the server concerning the particular signature. Following step 6.5 the retrieved condition is checked at step 6.6 to see if it is a "warn" condition, and if so processing proceeds to step 6.7, wherein a warning is displayed on the user computer's screen to warn the user about the received data. Following the display of the warning the particular process performed by the data filter program

Trend Micro, Inc.
EX1007-638

24 ends, although it should be noted that the data filter program will continue to run continuously in the background to check whenever data is received.

[0062] If the retrieved signatory condition is not a "warn" condition then processing proceeds to the evaluation of step 6.8 wherein the condition is checked to see if it is a "block" condition. If so then at step 6.9 the data filtering program 24 acts to block the receipt of the suspect data either by preventing the computer from downloading it from the server or from merely deleting it from the user computer's local storage, and again the user is preferably informed of this action. Following this action the processing performed by the data filter program 24 ends, although again the program will continue to run in the background to detect the receipt of new data.

[0063] If the evaluation of step 6.8 does not indicate that the retrieved signature is a "block condition", then the data filter program 24 processing ends, and the program returns to running in the background, as described previously.

[0064] The first embodiment of the invention therefore presents a computer security system whereby computer viruses and the like can be detected by individual users, who transmit warnings to a server which then broadcasts warnings as appropriate to all users if the number of individual warnings received from individual users exceeds certain thresholds. The use of thresholding in the server instils in the invention a degree of order, in that it is only once a particular threshold level of warnings have been received that action is taken by the server and user computers against the suspect data. This prevents, for example, inexperienced users who may mistake acceptable data for suspect data from initiating a security alert on the acceptable data which may prevent other users from receiving that data, for example.

[0065] Furthermore, the threshold levels for both the "warn" and "block" conditions may of course be set at any level required, and may even be simply one. The assumption is made, however, that in order for the embodiment to operate in a meaningful manner the warn threshold is less than or equal to the block threshold.

[0066] Within the preferred embodiment we have described the data filter program 24 as being located at each of the user computers 15 and running thereat. However, in a slight variation of the first embodiment it would of course be possible for the data filter program 24 to also run at the server to screen messages and data routed therethrough. In this case the operation of the data filter program would be exactly the same as previously described with respect to Figure 6, however it would mean that the user computers 15 would neither need the data filter program 24, the warning receipt program 26, or the signature and ID database 28, as all blocking operations could be performed centrally. However, in the case that it is desirable for users to be warned about data to be received, then it will still be necessary to have the data filter program 24, warning receipt program 26, and signature and ID database 28 at the user computer 15 as described previously in order to allow such warnings.

[0067] In another variation of the first embodiment, in another embodiment a step equivalent to step 4.2 can be performed at the user computers before a warning message is generated. Here, after a user has identified a piece of data as suspect and instructed her computer to send a warning message to the server, prior to compiling and sending the message the warning creation program 34 controls the computer to check if the data identified as suspect by the user could possibly be a virus. Such checking is substantially identical to the check performed at step 4.2, i.e. takes the form of determining the data type to see if it could feasibly be a virus. Viruses are frequently executable files, such as .exe or .bat on a PC, or macros such as are found in Microsoft® Word® or Excel®. Such a check can conveniently be performed by looking at any file extension on the data. If such checking does not indicate that the data could be a virus then no warning message is sent, as it is assumed that the user has simply made a mistake. Instead, if the data is of a data type which could be a virus (e.g. an executable or the like) then the warning message is transmitted.

[0068] The primary advantage of the first embodiment which makes use of the server to keep track of the number of warning messages received about a particular piece of suspect data is that the network traffic overhead introduced by the security system is minimised, as users need only send warning messages to the server, which can then broadcast warning condition messages as appropriate.

[0069] A second embodiment of the invention will now be described with reference to Figures 7 to 10.

[0070] The second embodiment of the invention presents a pure "peer to peer" system which does without the server 12 of the first embodiment. Instead, within the second embodiment each user computer 15 itself individually keeps count of the number of warning messages received from its peers concerning suspect data, and in turn when a user computer detects suspect data it generates and broadcasts a warning message to all of its peers. This arrangement has the advantage that it is not dependent upon a central server for operation, and is therefore much more robust and immune to the loss of one or more of its elements than the client-server architecture of the first embodiment.

[0071] Figure 7 illustrates the software elements present at a user computer 15 in the second embodiment of the invention. More particularly, as in the first embodiment the user computer 15 is provided with a computer readable storage medium 22 on which is stored a data filter program 24, a warning creation program 72, a warning receipt program 74, and a signature and ID database 76.

[0072] Within the second embodiment the data filter program 24 and the warning creation program 72 both

Trend Micro, Inc.
EX1007-639

continuously run in the background on the users computer at all times. The data filter program 24 operates in exactly the same manner as previously described in the first embodiment with respect to Figure 6, that is it acts to check the signature of received data against signatures of suspect data stored in the signature database 76, and takes action against any received suspect data as necessary. In this respect, the operation of the second embodiment is exactly the same as that of the first embodiment. The difference between the second embodiment and the first embodiment lies, however, in the fact that each user computer 15 directly broadcasts warnings to each of its peer computers (i.e. the other user computers 15), and in addition each user computer 15 also keeps its own track of the count of warning messages received against each piece of suspect data. These functions are performed by the warning creation program 72 and the warning receipt program 74, as described next.

[0073] The operation of the warning creation program 72 is shown in Figure 8. Here, at step 8.1 data is received at the user computer 15 over the network 10. At step 8.2 an evaluation is performed to determine whether the received data is a security threat or not. As with the first embodiment, this evaluation is preferably performed by the actual human user of the computer 15 who employs her expertise to determine whether or not the received data is a computer virus or the like. In alternative embodiments, however, it may be that this evaluation could be performed by a software agent installed on the users computer, the software agent acting to monitor the functions of the computer to determine if it is under attack from a computer virus. Within the first and second embodiments, however, we prefer the use of a human operator to perform the evaluation.

[0074] If the evaluation at step 8.2 concludes that the received data is not a threat then the processing performed by the warning creation program on the received data ends and the program returns to running in the background until further data is received in the future.

[0075] If it is determined at step 8.2 that the received data is a threat, then at step 8.3 a data signature for the received data is created. As in the first embodiment, the signature generation function may be a hash function or the like or any other suitable signature generation function which is capable of repeatably generating the same signature for a given set of input data. Furthermore, it should be noted here that each user computer 15 must use the same signature generation function, such that when the same set or piece of suspect data arrives at any of the user computers, each of them creates the same signature therefor.

[0076] At step 8.4 a check is made to see whether the generated data signature has already been stored in the signature and ID database 76, and if not then at step 8.8 the signature is stored in the database, and is allocated a globally unique ID. The ID for the signature is generated by the user computer, preferably from a list of ID's which have been specifically allocated to that user computer for allocation to signatures thereby.

[0077] Following step 8.8, at step 8.9 a message count is instantiated for the particular signature generated by the received data, and is set to the value one. As in the first embodiment, the various count values for each piece of suspect data may be stored in a one dimensional array indexed by signature ID.

[0078] Following step 8.9 at step 8.10 the warning creation program causes the user computer to broadcast the data signature and the signature ID to all of its peers as a warning message. As in the first embodiment, the warning message preferably takes a known format, with the signature and signature ID as particular fields within the warning message. Following the broadcast of the warning message, at step 8.10 processing of the warning creation program ends, and it returns to run in the background.

[0079] Returning to the evaluation at step 8.4, if it is determined that the data signature for the received data has already been stored (i.e. the data has already been identified as suspect by either the present user, or one of the other users who have then broadcast a warning message which has been received at each user computer) then processing proceeds to step 8.5, wherein the message count maintained for the particular signature is incremented to reflect the fact that the data has in fact been identified once again as a threat.

[0080] Next, at step 8.6 the signature ID is broadcast to every other user in the system as part of a warning message, and then following that at step 8.7 the appropriate warning or block conditions are set for the signature based on the warning and block threshold values. The specific operations required by step 8.7 are shown in more detail in Figure 10.

[0081] With reference to Figure 10, the steps required for setting the warning or block conditions for a particular data signature are as follows. At step 10.1 the message count for the particular data signature is retrieved, by indexing into the counter array using the signature ID. Then, at step 10.2 the retrieved message count is compared against the block threshold, and if larger than the threshold at step 10.3 the security condition for the data signature is set to "block". As in the first embodiment, this security condition is stored in the signature and ID database 76, and is used by the data filter program 24 to filter received data.

[0082] If the warning message count for the particular data signature is not greater than the block threshold then the message count is compared against the warning threshold. If it is not greater than the warning threshold then no warning or block condition is set for the particular data signature. In contrast, if it is greater than the warning threshold then the security condition for the particular data signature is set to "warn", which is again used by the filter data program 24 as described before in respect of the first embodiment.

[0083] Returning to Figure 8, once the warning or

Trend Micro, Inc.
EX1007-640

block conditions have been set for the data signature, the specific processing performed by the warning creation program 72 ends, and it returns to running in the background until further data is received.

[0084] Turning now to the operation of the warning receipt program 74, this program operates whenever a warning message generated by a warning creation program stored on another user's computer is received. The operation of the warning receipt program 74 is shown in Figure 9.

[0085] At step 9.1, a warning message is received from one of the other user computers. The warning message is of course in a predetermined format which identifies it as such. Then, at step 9.2 the warning receipt program 74 acts to parse the message to see if it contains a data signature relating to suspect data. Recall here that the data will have been determined as suspect by the user of the user computer from which the message was received. If it is determined that the message does not contain a signature, then this means that the data will have already been determined as suspect previously, and hence should already be stored in the user computer's signature and ID database 76. Therefore, at step 9.3 the warning receipt program acts to retrieve the message count for the particular suspect data to which the warning message relates, using the signature ID which will have been included in the warning message (recall here that at step 8.6 in the warning creation program where the suspect data has already been identified as such a warning message is still sent out including the suspect data signature ID). Then, at step 9.4 the retrieved message count for the suspect data is incremented to reflect the additional warning, and at step 9.5 the warning or block conditions for the data signature are also set, following the procedure of Figure 10, as described previously.

[0086] If it is determined at step 9.2 that the message does contain a signature, then processing proceeds to the evaluation of step 9.6 wherein it is determined whether or not the received signature has already been stored. If not processing proceeds to step 9.7 wherein the signature is stored in the signature and ID database 76 indexed by the signature ID included in the received warning message. At step 9.8 a message count is instantiated for the newly stored data signature, and is set to one. As described previously, the count values for each particular signature can be stored in a one dimensional array, indexed by each signature ID.

[0087] Following step 9.8 processing proceeds to step 9.5, wherein the warning or block conditions are set for the particular signature, using the procedure of Figure 10.

[0088] Returning to step 9.6, if it is determined that the signature received in the warning message has already been stored then processing proceeds to step 9.9, wherein the received signature ID is stored for use as another key to the signature. In such a case a count value should be instantiated related to the received signature ID, with its value set to the same value as the count indexed by the ID of the already stored signature which matches the received signature. Then, at step 9.10 both of the count values stored for each ID (the received ID, and the ID of the previously stored ID which matches the received signature) are incremented to reflect the receipt of the additional warning. Finally, following the increment of the message counts for the signature the warning and block conditions are set for the signature once more, using the procedure of Figure 10.

[0089] Therefore, as described the second embodiment provides a pure peer to peer system which incorporates the distributed warning capabilities of the security system of the present invention with the ability to threshold the warnings so as to be able to provide a graded reaction locally at each user computer. The provision of such a peer-to-peer operation provides for a more robust system, but as mentioned earlier might have detrimental network overheads due to the need to broadcast every warning message to every user.

[0090] As a modification to the operation of the second embodiment, an additional checking step analogous to step 4.2 of the first embodiment may be performed between steps 8.2 and 8.3 of Figure 8 depicting the operation of the warning creation program 72. Here, after the human user has identified data as suspect at step 8.2, an additional check is made to see if the data could possibly be a computer virus. Such checking preferably takes the form of determining the data type to see if it could feasibly be a virus. Viruses are frequently executable files, such as .exe or .bat on a PC, or macros such as are found in Microsoft® Word® or Excel®. Such a check can conveniently be performed by looking at any file extension on the data. If such checking does not indicate that the data could be a virus then processing ends, as it is assumed that the user has simply made a mistake. Instead, if the warning message is in fact a true warning and the suspect data could possibly be a virus, then processing proceeds to step 8.3, and onwards as previously described. The operation of the other programs in the second embodiment remains unchanged.

[0091] It should be noted here that as the second embodiment does not require a server to act centrally, but rather is a true peer-to-peer system, it is particularly suitable for implementation on networks such as ETHERNET and the like.

[0092] Various further modifications may be made to either of the first or second embodiments to produce further embodiments, as described next.

[0093] In further embodiments it is possible to add more sophistication to increase the power of the system. For example, users could be given the ability to respond to a warning to inform the server of whether they agree with the assessment or not - this would give the potential of reducing the count number if appropriate. The decision as to whether a particular program was a threat or not would then be a collective decision from the whole community.

Trend Micro, Inc.
EX1007-641

[0094] Furthermore it is also possible to build in trust models for each user such that some users have greater influence than others. Such trust models could be predefined according to the users position within the community/organisation, such that a warning received from a user with a high trust rating increments the signature count for a particular piece or set of suspect data more than another user with a lower trust rating. Each trust model may then also be adaptively varied by judging the accuracy of a prediction that a user has made against a final collective decision, and any further warnings registered by that user could be weighted according to the trust in that users demonstrated decision-making ability.

[0095] In other embodiments a community of organisations could be formed that would extend the collaboration beyond the limits of a single organisation through sharing of knowledge. Any discoveries in one organisation could be passed on to all other organisations in the community and vice versa. This would greatly extend the power of the system and in many cases allow rapid response to intrusions experienced in other organisations without any internal hits being experienced to the remaining organisations using the system.

[0096] Moreover, in another embodiment the actual number of warnings received concerning a particular piece or set of suspect data could be displayed to the user if the suspect data is received at the user's computer, and the user could then decide what action to take himself. Such a scheme is akin to the "warn" action of the primary embodiments, but provides the user with the additional information of how many other users have determined the data to be threat, thus providing the user with an assessment analogous to a community "confidence-level" in the data, such a level being of greater resolution than the simple "warn" or "block" conditions. Furthermore, such a scheme would not require the system to actually perform automatic blocking of suspect data, as this could be left to the user to do himself. This mode of operation provides the additional potential advantage that a highly skilled or confident user may not wish to act on the community warning if he is sure that it may be wrong, thus increasing the flexibility of the system.

[0097] Furthermore, in another embodiment based on the peer-to-peer architecture of the second embodiment, rather than broadcast to all peers e.g. all users in BT - sub-communities of peers are formed so that the initial broadcasting of warning messages concerning a particular is limited to a smaller number of peers. An example of such an architecture is shown in Figure 11, whence it will be seen that a plurality of sub-communities 111, 112, 113, and 114 may be provided, each connected to each other. Only when the count passes the warning threshold in one of the sub-communities is the suspect data signature distributed further. This limits the overheads incurred by false positives in that any particular sub-community would need to agree that a particular piece or set of data was a threat before it was broadcast to the wider community. "Crossover points" are provided between the sub-communities so that inter-group (or alternatively "inter-community") warnings may be passed on to other sub-communities by the provision of one of the peers 115 in a particular sub-community acting as the bridge via a communications link to another peer in a neighbouring sub-community, and only communicating with that other peer when the warning (or block) threshold has been passed in its own parent sub-community i.e. the sub-community has decided that the particular piece or set of suspect data really could be a threat.

[0098] The processing steps involved in such an embodiment are almost identical to those required in the second embodiment, with the addition that the processing steps depicted in Figure 10 are altered to include an additional step after step 10.3 or 10.5 wherein once the filter condition has changed (as a consequence of either step 10.3 or 10.5) if the computer is a bridging peer to another sub-community then an inter-group warning message is sent to the connected sub-community informing the other peer that a change in the filter condition has occurred for a particular piece or set of suspect data. Such an inter-group warning message should preferably include at least the data signature, as well as a warning condition indicator to communicate the threat level (e.g. warn or block) decided by the sending sub-community. By sending such a warning message to other sub-communities, the bridging peer is effectively declaring to the other sub-communities to which it is connected that its peers in its own sub-community have found the suspect data identified in the warning message to be a security threat.

[0099] Conversely, when an inter-group warning message is received at a bridging peer from another sub-community, the bridging peer should broadcast the inter-group warning message onwards to all its own peers as an intra-group message, to inform them of the threat assessment made by the neighbouring community. Each peer can then store the suspect data signature in its signature and ID database 76, for future reference by the data filter program 24 as needed (and in the same manner as already described in relation to the second embodiment).

[0100] Whether or not a user computer in a sub-community uses a forwarded inter-group warning message directly to initiate action (such as "warn" or "block" for the indicated suspect data) is optional, as it could be that it may wish to wait until it's own community has also agreed on action in respect of the same suspect data before actually initiating a "warn" or "block" condition for the data. Conversely, where a particular user-computer's own sub-community has already come to agreement on a particular piece or set of suspect data (i.e. either the "warn" or "block" threshold has been passed), then it may be that the user-computer may wish to wait until one or more connected sub-communities have also made the same finding before setting the appropriate

Trend Micro, Inc.
EX1007-642

condition in its database for the data. It will be apparent that there are a multitude of possibilities as to when any particular user computer may wish to actually set an action condition for a particular piece or set of suspect data. For example, an action condition may be set for a particular piece or set of suspect data only if a user computer's own sub-community has agreed on the action in addition to a threshold number of other sub-communities, or alternatively if merely a threshold number out of the total number of sub-communities has agreed on the action regardless of the particular user computer's own sub-community's finding. Other such alternatives representing possible combinations of the logical conditions required in order to set an action condition will be apparent to the intended reader.

[0101] It is clear that such an architecture of sub-communities is analogous to a plurality of inter-connected local area networks, although it is not essential that each peer in a particular sub-community need actually be located in the same actual LAN (although this may be preferable). This is because all that defines any particular sub-community is the common cause that upon detecting suspect data a user broadcasts a message to all of its peers in the same sub-community, and hence all that is required is some message transport mechanism between peers. Such a message transport mechanism may be a single LAN to which all the peers are connected, or may be a sequence of interconnected networks such as the Internet, with different peers in a sub-community actually being part of different networks.

[0102] The modification to the second embodiment to split the user computers into sub-communities provides the additional advantage over the basic second embodiment that it, at least partially, overcomes the primary disadvantage of the high signalling bandwidth which would be required for the transmission of the warning messages to every other peer, and hence allows for scalability to add more users. As an example, consider the case where a community consists of n peers, such that when a warning message is broadcast from one of them to each of the others $(n-1)$ individual intra-group warning messages are sent. Now, assume that in order to agree on action about a particular piece or set of data a proportion of $1/a$ peers have to agree i.e. the action threshold is n/a. In such a case a total of $n(n-1)/a$ intra-group warning messages have to be transmitted in total for any one particular piece or set of data. Thus, where there are 40 peers, and half of them have to agree before action can be taken, a total of 780 warning messages would have to be transmitted (without error) before the action is agreed upon.

[0103] Now consider that the community of n peers is split equally into x sub-communities, as shown in Figure 11. In this case, within a sub-community for any single warning from one peer a total of $(n/x - 1)$ intra-group warning messages are broadcast. Assuming that in order to agree on action about a particular piece or set of data a proportion of 1/a peers have to agree i.e. the action threshold is n/ax, then a total of $n(n-x)/ax^2$ intra-group warning messages have to be broadcast within a single sub-community for action to be agreed on a single piece of suspect data. Assuming that each sub-community has to reach agreement within itself and communicate to each other group that it has so agreed before action can be taken by any peer in any sub-community, there will still only be a total of $n(n-x)/ax$ intra-group warning messages broadcast in total (the sum of the number of warning messages in total broadcast in each sub-community), plus $x(x-1)$ inter-group messages between sub-communities, plus $(n/x - 1)$ further intra-group messages from the bridging peer in each sub-community to each other peer in its respective community informing them of the findings of other sub-communities (i.e. forwarding received inter-group messages onwards). Note that with respect to this latter number of intra-group messages, $(n/x - 1)$ messages are sent from a bridging peer in a sub-community for each inter-group warning message received from another sub-community, such that where full knowledge of every other sub-community's findings about a particular piece or set of suspect data is required to be disseminated throughout a particular sub-community, the total number of intra-group messages transmitted by a single bridging node forwarding on the findings of other sub-communities would be $(x-1)(n/x - 1)$. This leads to the total number of intra-group messages forwarding inter-group messages across the entire community of users of $x(x-1)(n/x-1)$.

[0104] For example, if there are 40 peers split into 2 sub-communities of 20, then in each sub-community 190 messages are sent before action is agreed (i.e. the action threshold is reached) in each sub-community, 2 messages are exchanged between the respective bridging peers of each sub-community, one in each direction, and then the respective bridging peers each send 19 messages to inform their own peers of the finding of the other sub-community, to give a total of 420 messages. This is substantially less than the 780 messages which were required when the peers were not split into sub-communities.

[0105] It should be apparent that the above numeric examples are dependent solely on the logical conditions which were ascribed to the group operation, and in particular in the example above the need for every sub-community to agree on action about a particular piece or set of data and each issue its own inter-group warning message before any action is taken. Where slightly lower thresholds are set for action (e.g. if only half the number of sub-communities are required to agree), then the number of both inter- and intra-group warning messages which would be required to be broadcast before action is agreed would be further reduced.

[0106] In any of the embodiments of the invention described above and in other embodiments not described it should be noted that the one or more "threshold values" against which the respective counts for pieces or sets of suspect data are compared may be any value,

Trend Micro, Inc.
EX1007-643

and may be fixed or dynamically adaptive. Furthermore the actual values may be generated by any appropriate function, such as for example by a random number generator, or as a function of some metric of accuracy of users in identifying viruses. Furthermore, it is foreseen that the threshold values may not simply be actual numbers which the respective counts must be greater than, but could instead be probabilistic in nature. Furthermore, various logical functions or rules can be used for their creation, including for example, fuzzy logic. The threshold values may therefore be any values, derived by any function or method, and are not limited to the particular examples of threshold values given in the specific embodiments described herein.

[0107] As described, the present invention therefore provides a collaborative computer security system wherein a distributed decision is taken involving all or substantially all of the users of the system, to effectively allow the users to "vote" on whether or not a particular piece or set of suspect data is a threat. By providing such a distributed system the response time to a computer virus attack can be improved over the prior art case where a central authority is required to perform analysis and take action, thereby reducing the potential damage which may be caused by such an attack.

Claims

1. A computer security system for use in a network environment comprising at least a first group of user computers arranged to communicate over a network, the system comprising a warning message exchange system operable to allow the communication from the user computers of warning messages relating to suspect data identified as a possible security threat; a message counting system operable to maintain a count for every particular piece or set of suspect data based on the number of warning messages communicated relating thereto; and network security means operable to act in respect of any particular piece or set of suspect data for which the count maintained therefor is substantially equal to or greater than at least one threshold value.

2. A computer security system according to claim 1, wherein the judgement as to whether any piece or set of data present at any one or more of the user computers is a possible threat is performed by the respective human user or users of the or each user computer.

3. A computer security system according to any of the preceding claims, wherein the network security means is further operable to compare the maintained count for a particular piece or set of suspect data against a plurality of different thresholds, and to take different action in respect of the data dependent upon which threshold or thresholds are met.

4. A computer security system according to any of the preceding claims, wherein the action taken by the network security means comprises warning each of the users as to the suspect nature of a particular piece or set of data.

5. A computer security system according to any of the preceding claims, wherein the action taken by the network security means comprises preventing the transmission of the particular piece or set of suspect data across the network.

6. A computer security system according to any of the preceding claims, wherein the or each warning message comprises at least an identifier of the piece of or set of suspect data to which it relates.

7. A computer security system according to claim 6, wherein the identifier is the piece or set of suspect data itself.

8. A computer security system according to claim 6, wherein the identifier is a repeatably generatable signature substantially unique to the piece or set of data.

9. A computer security system according to any of the preceding claims, and further comprising a network server arranged to receive the or each warning message communicated from the user computers, and further arranged to host the message counting system and at least part of the network security system, wherein that part of the network security system hosted by the server is operable to determine the action which should be taken against a particular piece or set of suspect data, and to communicate an action condition indicator indicative of the determined action to each of the plurality of users.

10. A computer security system according to any of the preceding claims, wherein a warning message generated by a user computer is broadcast to every other user computer, and each user computer is further arranged to host its own message counting system and network security system operable in accordance with any of the preceding claims.

11. A computer security system according to any of the preceding claims, and further comprising an inter-group communications system operable to allow the communication of warning messages relating to suspect data identified as a possible security threat from the first group of user computers to one or more other groups of other user computers.

Trend Micro, Inc.
EX1007-644

12. A computer security system according to claim 11, wherein the inter group warning messages are transmitted by the inter-group communications system if the number of warning messages communicated by the user computers in the first group concerning a particular piece or set of data is substantially equal to or greater than at least one threshold value.

13. A computer security system according to claim 12, wherein the threshold value or values is/are the same as those at which the network security system takes action.

14. A computer security system according to any one of claims 11, 12, or 13, wherein the inter-group communications system is further arranged to receive inter-group warning messages from the one or more other groups of user computers.

15. A computer security system according to claim 14, and further comprising a second message counting system operable to maintain a count for every particular piece or set of suspect data based on the number of inter-group warning messages communicated relating thereto; and wherein the network security means is further operable to act against any particular piece or set of suspect data for which the count maintained therefor by the second message counting system is substantially equal to or greater than at least one inter-group threshold value.

16. A computer security system according to claim 15, wherein the network security means is further arranged to act against a particular piece or set of suspect data only if both the respective counts maintained by the message counting system and the second message counting system are substantially equal to or greater than at least the threshold value and the inter-group threshold value respectively.

17. A computer security system according to any of the preceding claims, wherein the message counting system is further arranged to store one or more weighting coefficients relating to one or more particular user computers, and to increment the count maintained for a particular piece or set of suspect data by an amount based upon the weighting coefficient when a warning is received from the one or more particular user computers relating to the particular piece or set of suspect data.

18. A computer readable storage medium storing one or more computer programs which when run on a computer causes the computer to operate in the manner of a system according to any of the preceding claims.

Trend Micro, Inc.
EX1007-645

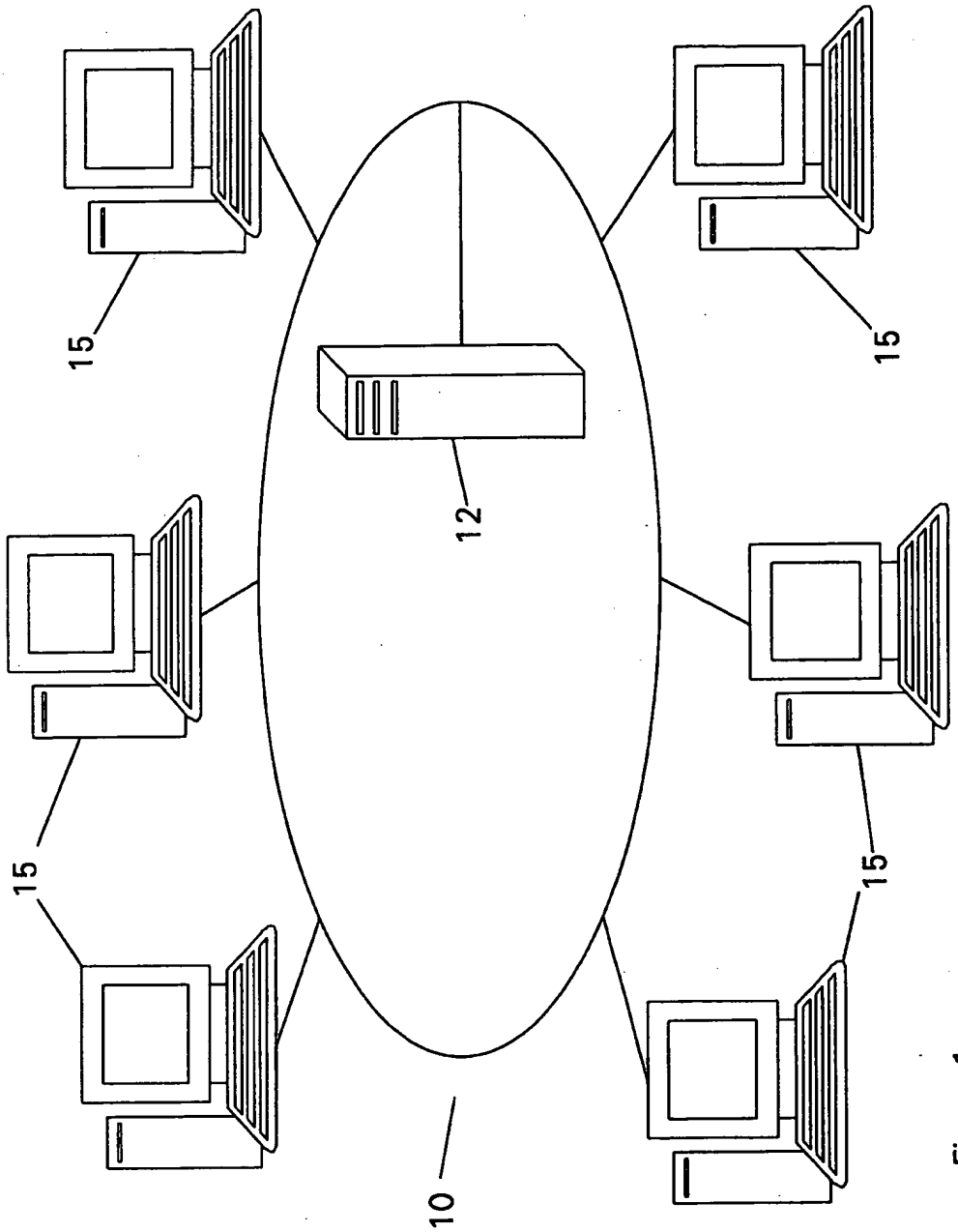


Figure 1

Trend Micro, Inc.
EX1007-646

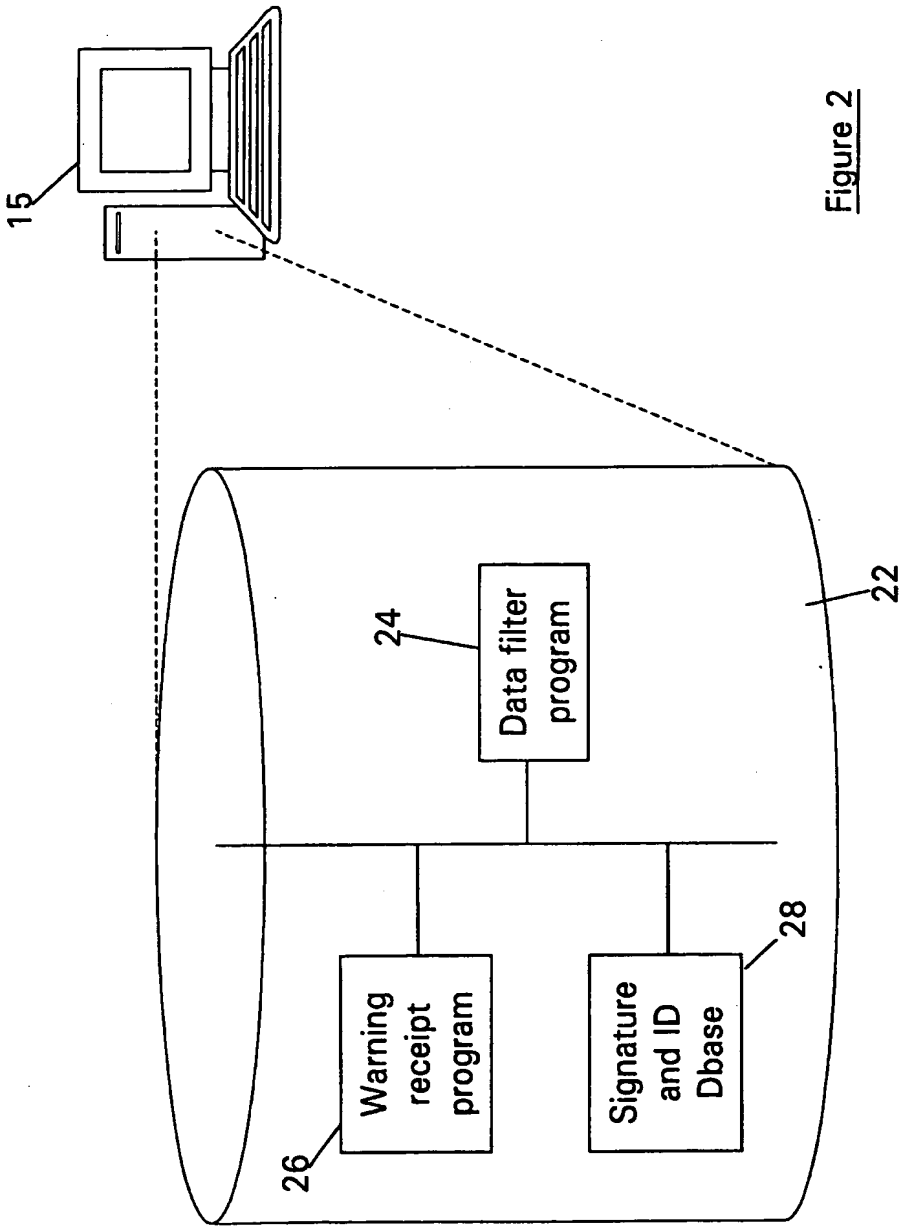


Figure 2

Trend Micro, Inc.
EX1007-647

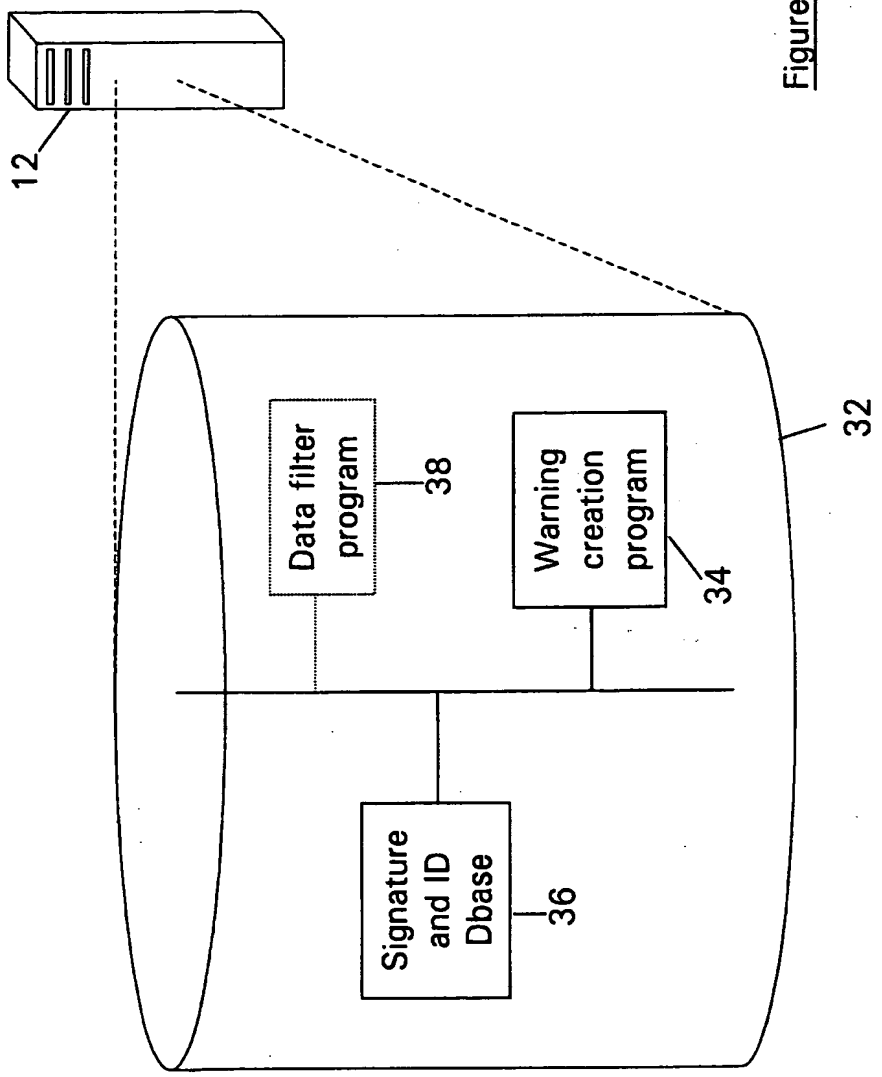


Figure 3

Trend Micro, Inc.
EX1007-648

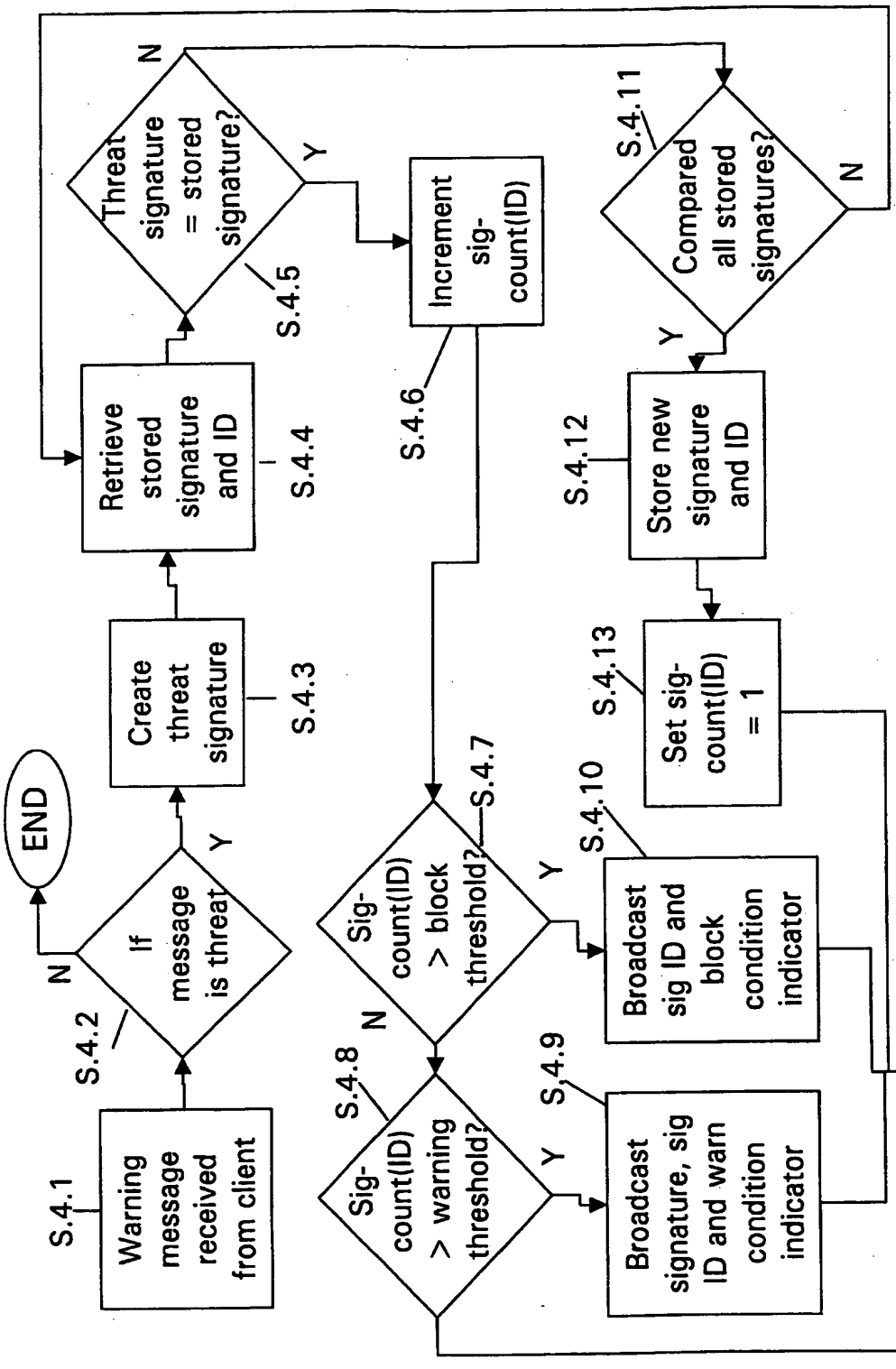


Figure 4

Trend Micro, Inc.
EX1007-649

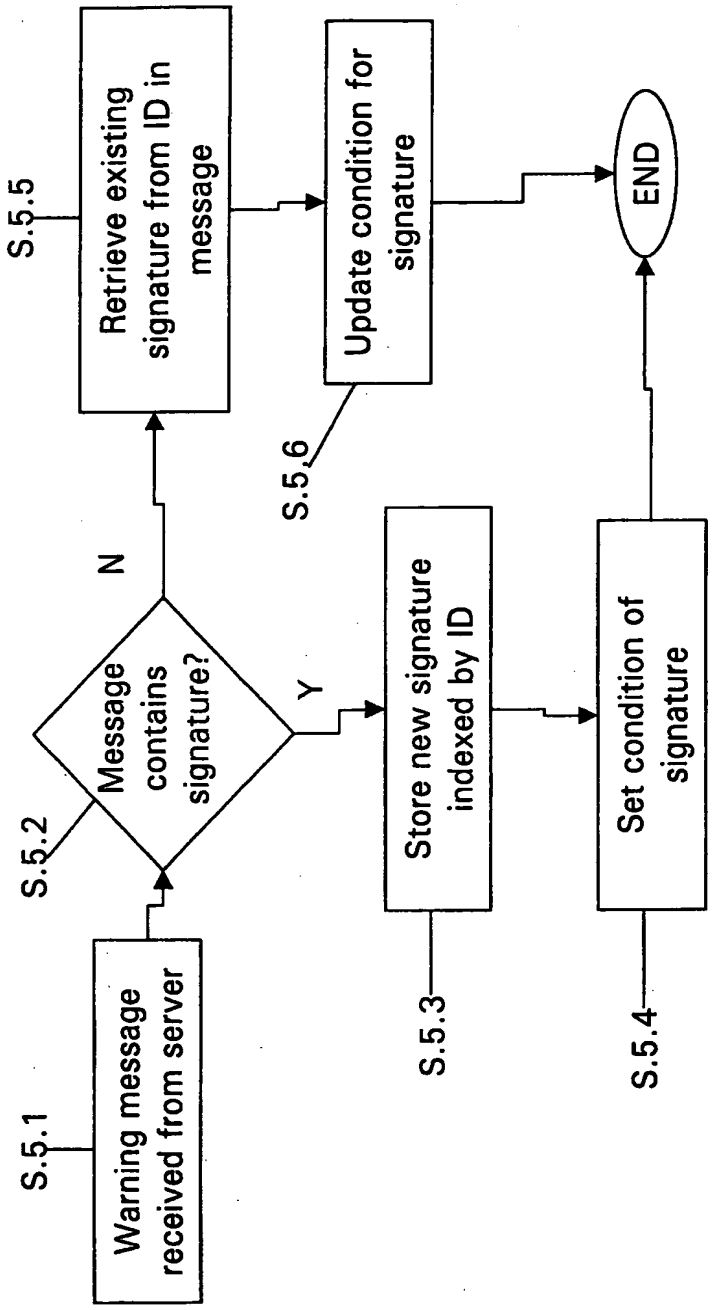


Figure 5

Trend Micro, Inc.
EX1007-650

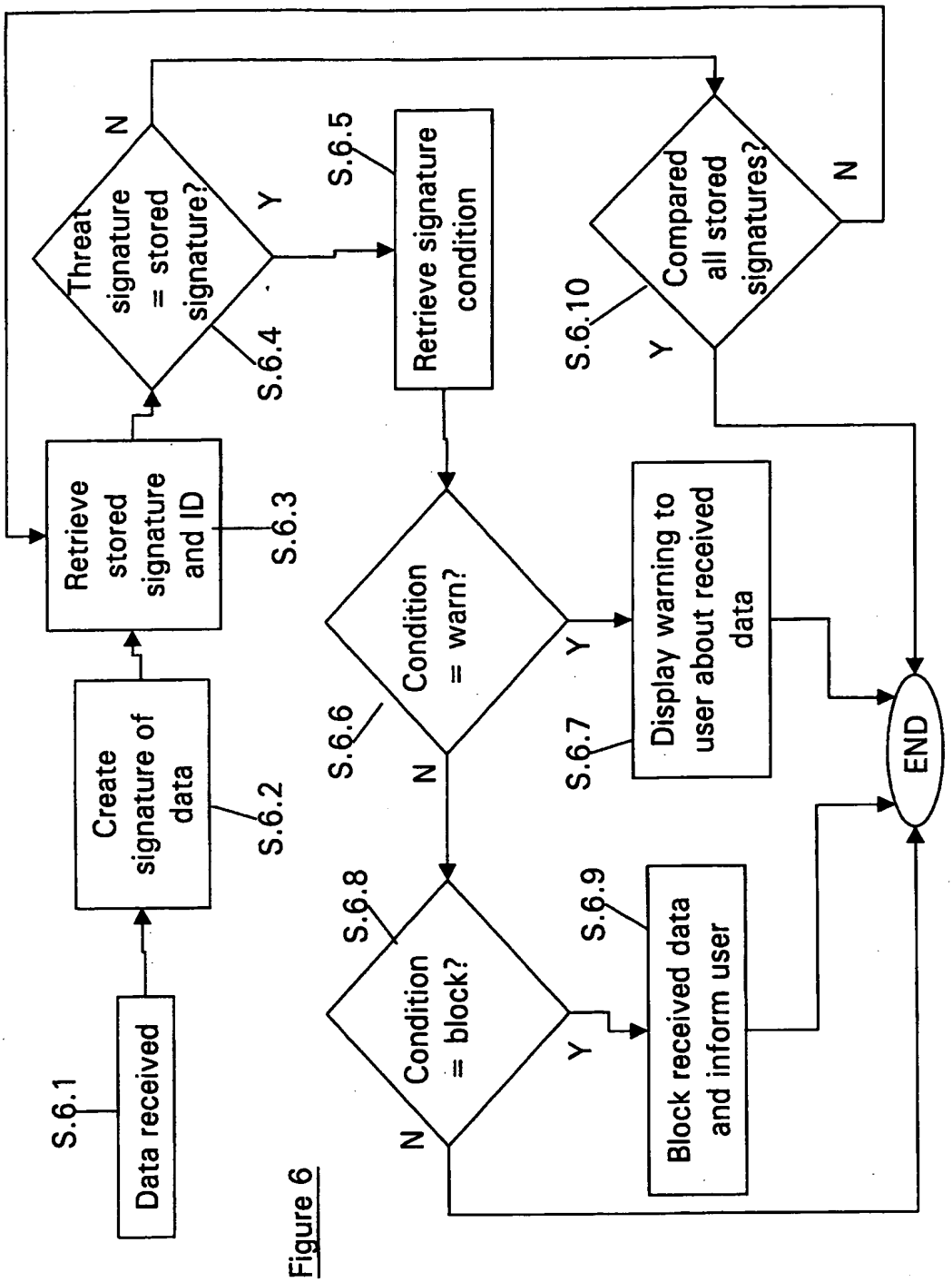


Figure 6

Trend Micro, Inc.
EX1007-651

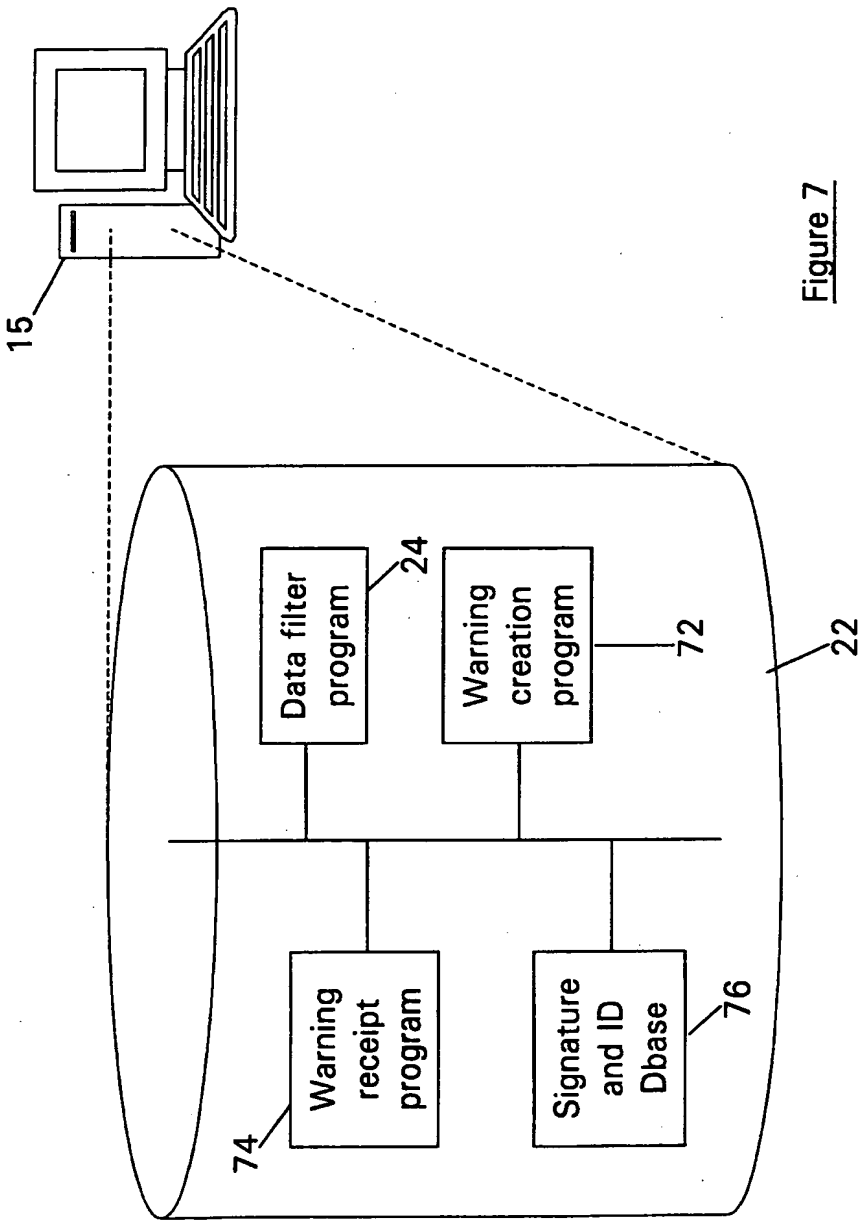


Figure 7

Trend Micro, Inc.
EX1007-652

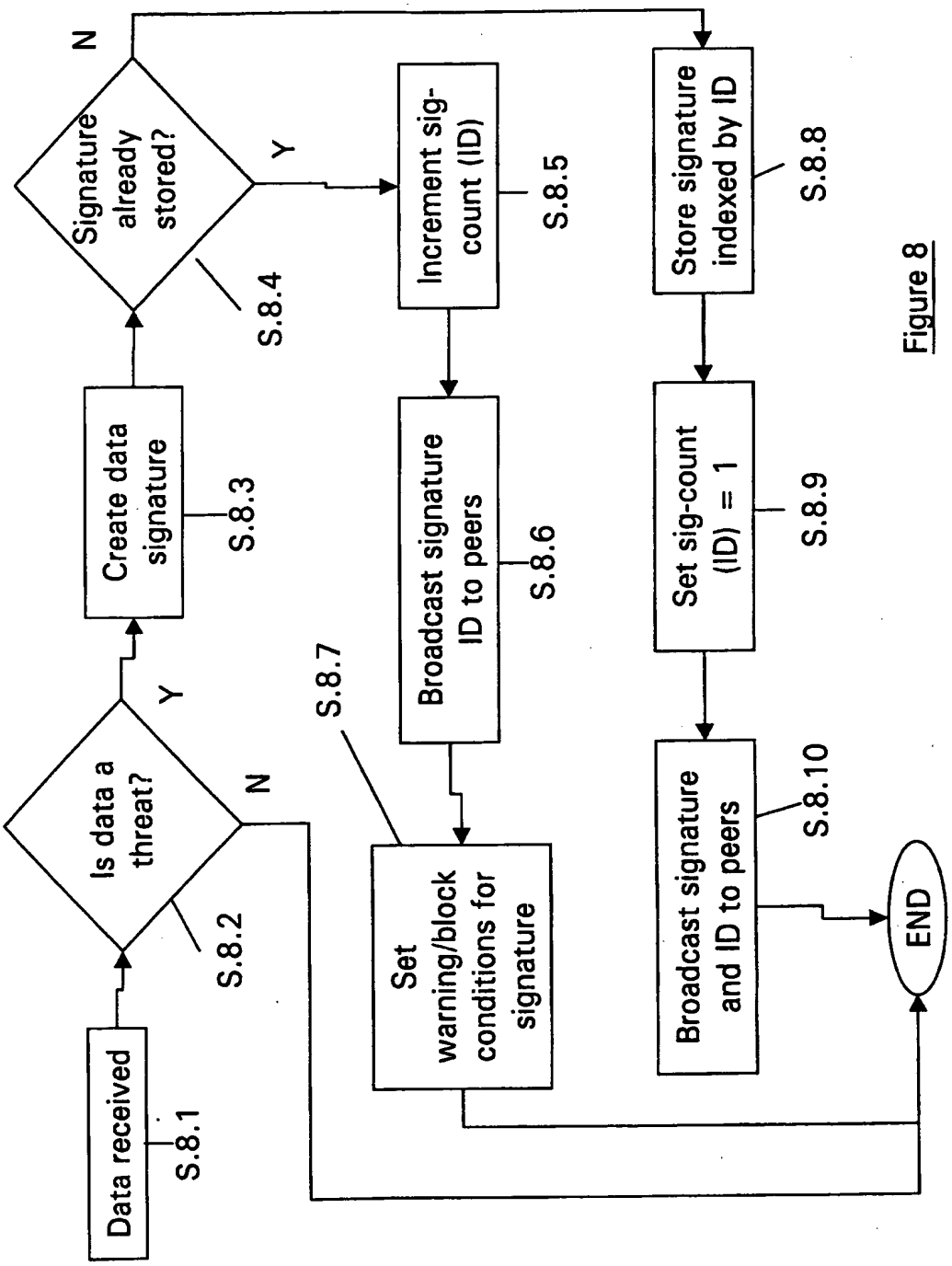


Figure 8

Trend Micro, Inc.
EX1007-653

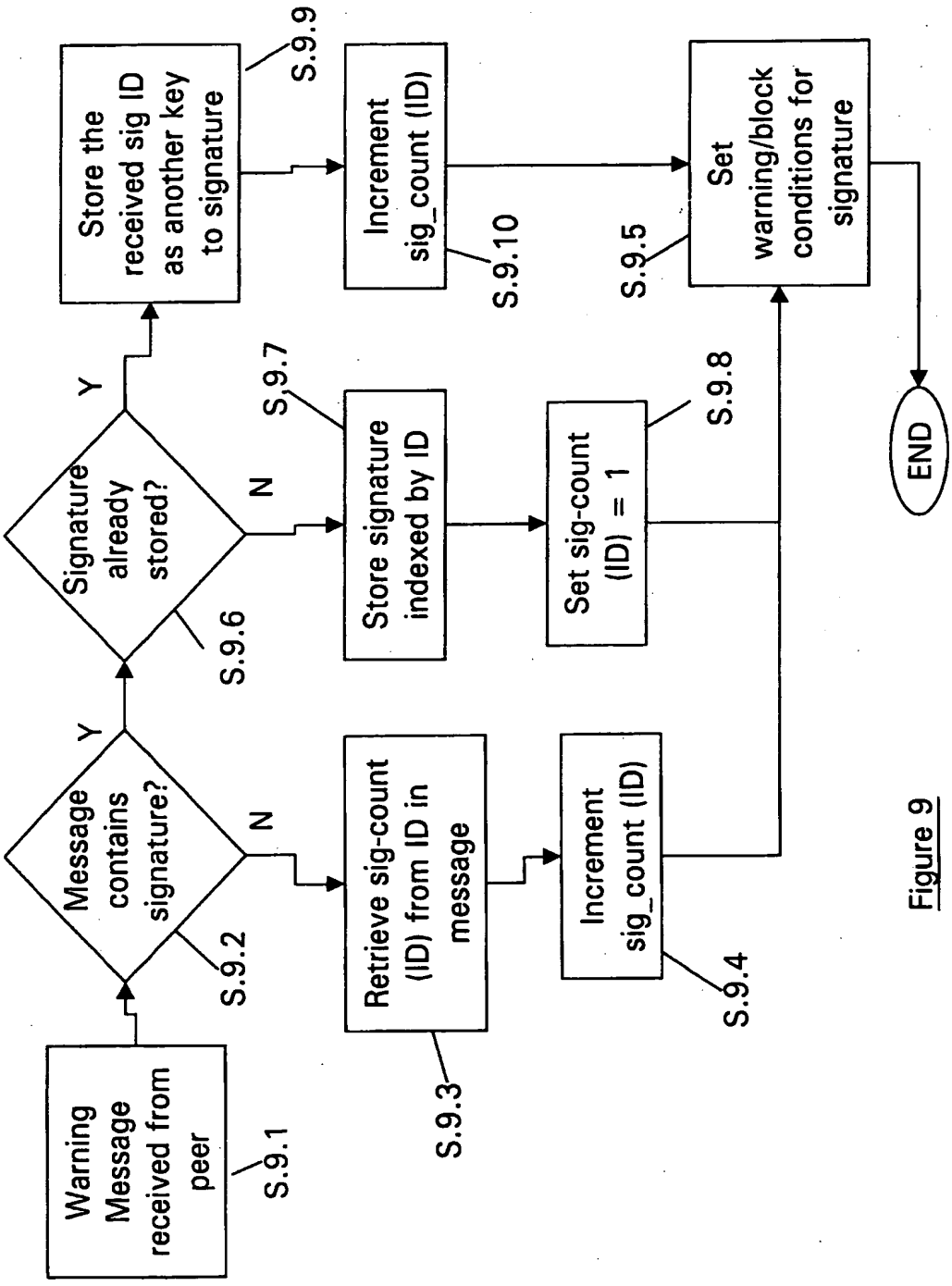


Figure 9

Trend Micro, Inc.
EX1007-654

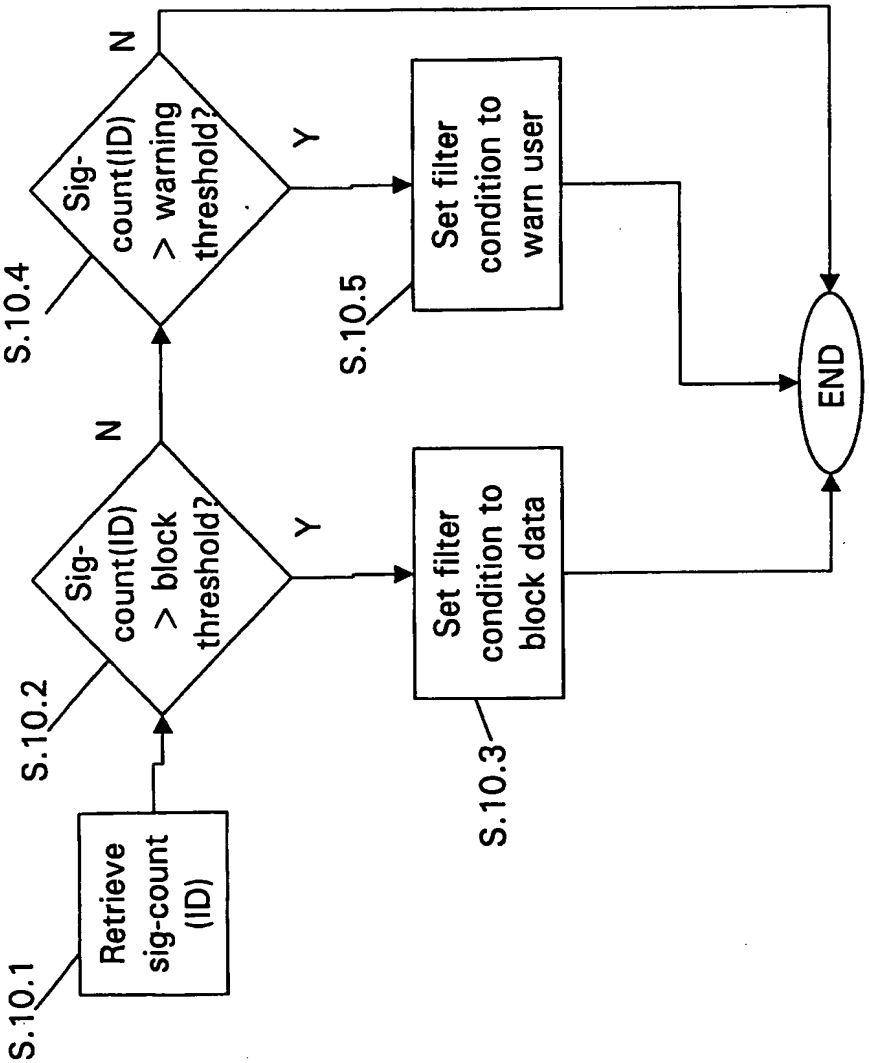


Figure 10

Trend Micro, Inc.
EX1007-655

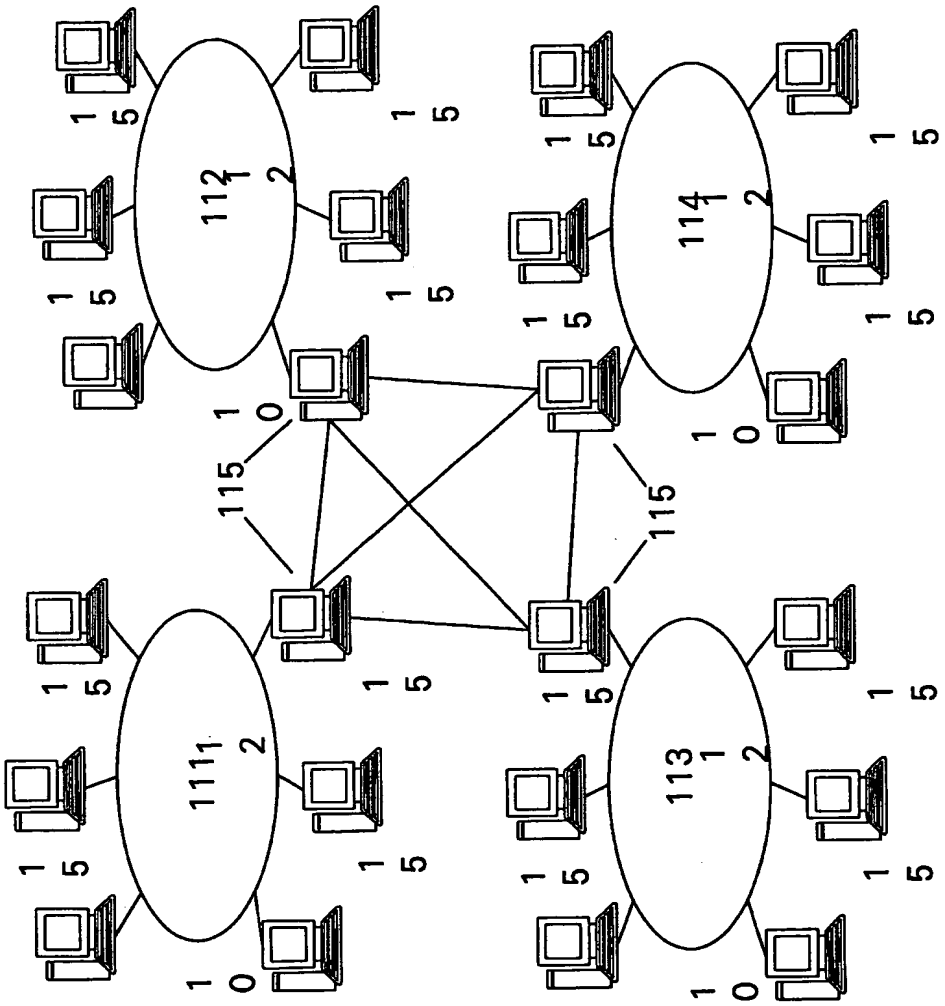


Figure 11

Trend Micro, Inc.
EX1007-656

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 01 30 9795

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
X	US 5 440 723 A (KEPHART JEFFREY O ET AL) 8 August 1995 (1995-08-08) * abstract; figures 2,3,7,8 * * column 4, line 29 - column 5, last line * * column 19, line 45 - column 20, last line *	1	G06F1/00 H04L29/06
A		2,3,5-8, 10-14	
A	US 5 621 889 A (EMERY THIERRY ET AL) 15 April 1997 (1997-04-15) * abstract; figure 2 * * column 2, line 9 - line 55 * * column 3, line 30 - column 4, line 43 *	1-3,17	
A	WO 01 71499 A (COUNTERPANE INTERNET SECURITY) 27 September 2001 (2001-09-27)		
A	ZENKIN D: "Fighting Against the Invisible Enemy - Methods for detecting an unknown virus" COMPUTERS & SECURITY. INTERNATIONAL JOURNAL DEVOTED TO THE STUDY OF TECHNICAL AND FINANCIAL ASPECTS OF COMPUTER SECURITY, ELSEVIER SCIENCE PUBLISHERS. AMSTERDAM, NL, vol. 20, no. 4, 31 July 2001 (2001-07-31), pages 316-321, XP004254268 ISSN: 0167-4048		TECHNICAL FIELDS SEARCHED (Int.Cl.7) G06F H04L

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	31 July 2002	Powell, D

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

Trend Micro, Inc.
EX1007-657

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 01 30 9795

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

31-07-2002

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5440723 A	08-08-1995	NONE	
US 5621889 A	15-04-1997	FR 2706652 A1	23-12-1994
		CA 2125445 A1	10-12-1994
		DE 69428631 D1	22-11-2001
		EP 0629940 A1	21-12-1994
		ES 2164091 T3	16-02-2002
		NO 942108 A	12-12-1994
WO 0171499 A	27-09-2001	US 2002087882 A1	04-07-2002
		AU 4735101 A	03-10-2001
		WO 0171499 A1	27-09-2001

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

Trend Micro, Inc.
EX1007-658

PTO/SB/06 (12-04)
Approved for use through 7/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number: 11/477807

APPLICATION AS FILED – PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	SMALL ENTITY RATE ($)	SMALL ENTITY FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY FEE ($)
BASIC FEE (37 CFR 1.16(a), (b), or (c))							300
SEARCH FEE (37 CFR 1.16(k), (l), or (m))							500
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))							200
TOTAL CLAIMS (37 CFR 1.16(i))	47 minus 20 =	* 27	x =		OR	x 50 =	1350
INDEPENDENT CLAIMS (37 CFR 1.16(h))	8 minus 3 =	* 5	x =		OR	x 200 =	1000
APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).					250 / 300	
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL			TOTAL	3350

APPLICATION AS AMENDED – PART II

10-31-06

AMENDMENT A	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE ($)	SMALL ENTITY ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	* 47	Minus	** 47	=	x =		OR	x 50 =	
Independent (37 CFR 1.16(h))	* 8	Minus	*** 8	=	x =		OR	x 200 =	
Application Size Fee (37 CFR 1.16(s))							OR	300	
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

AMENDMENT B	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)	OR	RATE ($)	ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	*	Minus	**	=	x =		OR	x =	
Independent (37 CFR 1.16(h))	*	Minus	***	=	x =		OR	x =	
Application Size Fee (37 CFR 1.16(s))							OR		
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2

OIPE IAP44
OCT 31 2006
PATENT & TRADEMARK OFFICE

Attorney Docket: 011765-0355986
Client Reference: AF/JG/P10717US

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re PATENT APPLICATION of: MELVYN MORRIS , ET AL.
Application No.: 11/477,807

Confirmation Number: 4785

Group Art Unit: 2131

Filed: June 30, 2006

Examiner: Unassigned

Title: METHODS AND APPARATUS FOR DEALING WITH MALWARE

PRELIMINARY AMENDMENT

Commissioner for Patents
P.O. Box 1450
Alexandria,VA 22313-1450

Sir:

Prior to examination of the above-identified application, please amend the application as outlined herein. There are no amendments to the claims.

Trend Micro, Inc.
EX1007-660

IN THE SPECIFICATION:

Page 3, please amend the first paragraph as indicated below:

In the system of ~~WO-A-2004/0083408~~ US-A-2004/0083408, a worm in a file is detected by examining connection attempts made by the specific file running on a computer.

REMARKS

The foregoing amendment to the specification corrects an inadvertent typographical error in the application country designation code of a cited patent reference. The country code "WO" has been amended to correctly reflect the correct country code for the application referenced, i.e., "US". No new matter is added, and entry of the amendment is therefore respectfully requested and deemed in order.

Please charge any fees associated with the submission of this paper to Deposit Account Number 033975. The Commissioner for Patents is also authorized to credit any over payments to the above-referenced Deposit Account.

Respectfully submitted,

PILLSBURY WINTHROP SHAW PITTMAN LLP
CHRISTINE H. MCCARTHY
Reg. No. 41844
Tel. No. 703.770.7743
Fax No.

Date: October 31, 2006
P.O. Box 10500
McLean, VA 22102
(703) 770-7900

IAP44
OCT 31 2006
PATENT & TRADEMARK OFFICE

Docket Number: 011765-0355986
Client Reference: AF/JG/P10717US

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re the Application of

MELVYN MORRIS, et al.

Group Art Unit: 2131

Application No.: 11/477,807

Examiner:

Filed: June 30, 2006

Confirmation No.: 4785

For: METHODS AND APPARATUS FOR DEALING WITH MALWARE

INFORMATION DISCLOSURE STATEMENT

Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

Sir:

Pursuant to 37 CFR 1.56, the attention of the Patent and Trademark Office is hereby directed to the reference(s) listed on the attached PTO-SB/08b. Copies of the references are provided herewith. It is respectfully requested that the information be expressly considered during the prosecution of this application, and that the references be made of record therein and appear among the "References Cited" on any patent to issue therefrom. Applicants respectfully request the Examiner return an initialed copy of the enclosed Form PTO-SB/08b to Applicants with the next Office communication to indicate that the references have been considered, per MPEP § 609.

This Information Disclosure Statement is being filed before the mailing date of the first Office Action on the merits in the present application. No certification or fee is required.

Respectfully Submitted,
PILLSBURY WINTHROP SHAW PITTMAN LLP

Christine H. McCarthy
Registration Number 41844
Customer Number 00909

Date: October 30, 2006
Telephone: (703) 770-7900
Facsimile: (703) 770-7901
P.O. Box 10500
McLean, VA 22102

Trend Micro, Inc.
EX1007-663

OIPE IAP44
OCT 31 2006
PATENT & TRADEMARK OFFICE

PTO/SB/08a (07-05)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Substitute for form 1449A/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 1 of 1

Complete if Known	
Application Number	11/477,807
Filing Date	06/30/2006
First Named Inventor	MELVYN MORRIS
Art Unit	2131
Examiner Name	
Attorney Docket Number	011765-0355986

U. S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document Country Code[3] Number[4] Kind Code[5] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T[6]
		WO 02/33525 A2	04/25/2002			
		WO 03/021402 A2	03/13/2003			
		WO 2004/097602 A2	11/11/2004			
		EP 1 280 040 A3	03/24/2004			
		EP 1 549 012 A1	06/29/2005			

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
25 April 2002 (25.04.2002)

PCT

(10) International Publication Number
WO 02/33525 A2

(51) International Patent Classification[7]: G06F 1/00

(21) International Application Number: PCT/SG01/00213

(22) International Filing Date: 17 October 2001 (17.10.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
200005973-3 17 October 2000 (17.10.2000) SG

(71) Applicant and
(72) Inventor: CHUANG, Shyne-Song [SG/SG]; Singapore Post Centre, P.O. Box 493, Singapore 914017 (SG).

(74) Agent: DREW & NAPIER LLC; 20 Raffles Place, #17-00 Ocean Towers, Singapore 048620 (SG).

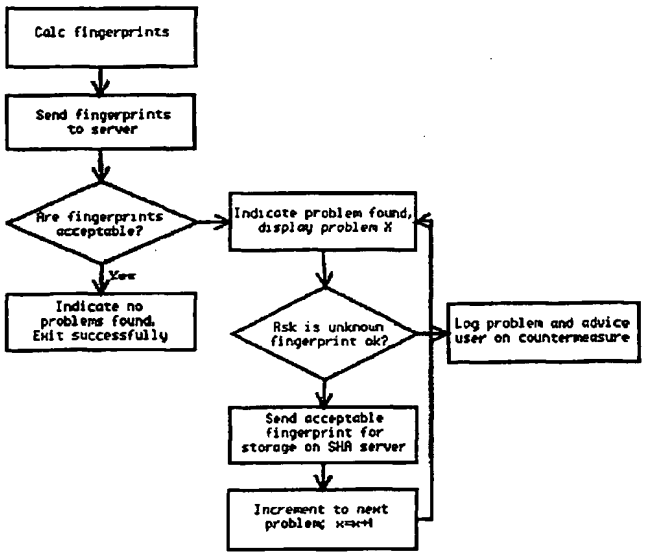
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: A METHOD AND SYSTEM FOR DETECTING ROGUE SOFTWARE



(57) Abstract: A method of detecting rogue software includes the step of creating a first database containing pre-calculated fingerprints for each file relating to typical operating systems and application software, wherein the pre-calculated fingerprints are calculated using one or more cryptographic formulae. The one or more cryptographic formulae are then used to calculate fingerprints of files on a computer system which is to be scanned for rogue software. The fingerprints calculated for the files on the computer system are compared with the fingerprints which are contained in the first database of pre-calculated fingerprints. Files on the computer system which may contain rogue software are identified by identifying files the calculated fingerprints of which do not correspond to the pre-calculated fingerprints which are stored in the first database.

WO 02/33525 A2

BEST AVAILABLE COPY

Trend Micro, Inc.
EX1007-665

A METHOD AND SYSTEM FOR DETECTING ROGUE SOFTWARE

Field of the invention

The present invention relates to a method and system for detecting rogue software such as trojan horses, root-kits, viruses and other unauthorized software which masquerades as valid software) on a computer system or data processing device such as a personal digital assistant. It relates particularly but not exclusively to a method and system for calculating and comparing fingerprints for files which are used either on a stand-alone computer system or on a computer system which is part of a computer network.

Back-ground to the Invention

Undesired rogue software is a nuisance and security threat. As computer systems and other information devices become even more interconnected with modern day networking technology and the Internet, the danger from rogue software has magnified considerably. Instead of being programmed to do damage once, today's rogue software can continue to receive commands and do the bidding of an unauthorized intruder for an extended period of time, effectively giving the creator of the rogue software continuous illegal access to a computer system.

One example of rogue software is the so-called trojan horse. Such software may be installed by innocent users unknowingly (whether via social engineering or otherwise) or it may be installed by an attacker when a system has been broken into. These trojan horses are back doors which allow an attacker to reconnect back into the compromised system and illegally access files and make unauthorized changes.

A trojan horse typically consists of new software and has new functionality. It is installed on a compromised system and disguised to look like original system software whenever necessary, so as to avoid detection. Sometimes, the trojan

horse is a modified piece of original system software and is almost identical to the one it replaced. However, other techniques are also used to obfuscate its existence.

Once the trojan horse is installed, a user continues operating his/her system without knowing that an intruder is now able to access his/her data illegally whilst remaining hidden. These are highly relevant problems which are encountered on a day to day basis. Trojan horses such as "Back Orifice" or "Netbus" hit PC systems in the late 1990s, and "root-kits" are a concern for various UNIX systems.

Normally, according to current practice, little is done to prevent or detect such rogue software. Anti-virus vendors maintain databases of rogue software signatures, and their software searches for files on a system associated with all known rogue software. Unfortunately, this technique has inherent scaling problems - the more signatures there are, the slower the scan process for each file. More importantly, the only rogue software types that can be detected are the ones that the anti-virus vendors know about. If the rogue software is, as an example, a custom trojan horse built by an expert professional hacker for penetrating a specific target, none of the anti-virus vendors will know about it. Therefore none of the anti-virus tools will be looking for it and the attacker and their trojan horse will exist completely undetected.

A more recent incremental innovation with this technology involves smarter scanning engines. Aside from looking for signatures of known rogue software, they are also able to look for software code that appears to be doing unusual things. This allows the scanning engine to detect additional rogue software that may not be known and whose signatures may not be in the database. However, this approach also has limitations. Trojan horses can be encrypted or compressed using special proprietary algorithms or encryption keying material. The rogue software is shipped in an encrypted and/or compressed format where it appears to be gibberish to a scanner. This rogue software is then decompressed or decrypted upon execution on the victim's computer system. A single trojan horse can thus be encrypted or compressed into thousands of

possibilities, each with its own unique signature. Traditional scanning technology will fail miserably when attempting to detect this type of rogue software, since there is no way that anti-virus engineers can keep track of thousands of mutations of the same piece of rogue software.

To summarise, today's scanning techniques for detecting rogue software fail for two main reasons:

1. They cannot detect unknown rogue software that has not already been identified. This is a serious problem because it is this kind of rogue software that may involve professional hackers and therefore warrant serious attention.
2. They cannot efficiently detect rogue software which has mutated, using new methods of compression and/or encryption. This problem exists to a large extent even for rogue software that is already known.

Another approach for detecting rogue software is to ensure that a system's files have not been altered, rather than looking for signatures of rogue software. If a system has no added files and all files remain unchanged from their original, unaltered state, it is clear that no rogue software is present on the system.

Academic work at Purdue University by Gene Kim and Gene Spafford resulted in a product called Tripwire which is now commercially sold. The product requires users to generate a database containing fingerprints of files on a system when the system is freshly loaded and in a pristine state. Subsequently, fingerprints can be recalculated and compared with the database of original pristine fingerprints and detect changes which have been made to the computer system.

This technology requires users to generate a database of fingerprints of a system's files while it is still pristine and free from alteration. This is not always feasible because many systems would already have been placed on public networks and exposed to risk for some time (often years). Since changes can be detected only by calculating new fingerprints and comparing them with the

database of original fingerprints, any rogue software which already exists when the original fingerprint database was generated will not be detected.

Existing products do not use a central database of fingerprints which are acceptable for a broad collection of system and application software. Therefore users need to make the following tedious and expensive steps when installing a software upgrade:

1. Schedule downtime in which to create the new database of fingerprints;
2. Re-calculate fingerprints to ensure that no rogue software has been added;
3. Install the software upgrade; and
4. Generate a new fingerprint database.

This is often a time-consuming and costly exercise.

Summary Of The Invention

Therefore it is an object of the present invention to provide a more reliable and effective method of identifying rogue software on a computer system or device, especially rogue software with unknown signatures or characteristics. The invention is preferably usable on systems or devices that have already been exposed to risk of intrusion by rogue software, and in cases where no fingerprints for the files on the system were calculated or archived when the system or device was known to be in a pristine state.

According to a first aspect of the invention, there is provided a method for detecting rogue software including the steps of:

(a) creating a first database containing pre-calculated fingerprints for each file relating to typical operating systems and application software, wherein the pre-calculated fingerprints are calculated using one or more cryptographic formulae;

(b) using the one or more cryptographic formulae to calculate fingerprints of files on a computer system which is to be scanned for rogue software;

(c) comparing the fingerprints calculated for the files on the computer system with the fingerprints which are contained in the first database of pre-calculated fingerprints; and

(d) identifying files on the computer system which may contain rogue software by identifying files the calculated fingerprints of which do not correspond to the pre-calculated fingerprints which are stored in the first database.

According to a second aspect of the invention, there is provided a system for detecting rogue software including:

(a) a first database containing pre-calculated fingerprints for each file relating to typical operating systems and application software, the fingerprints having been calculated using one or more cryptographic formulae;

(b) a software component which uses one or more cryptographic formulae to calculate fingerprints for files on a computer system; and

(c) a software component which compares the calculated fingerprints for the files on the computer system with corresponding pre-calculated fingerprints stored in the first database, such that files on the computer system which may contain rogue software are identified by identifying files the calculated fingerprints of which do not correspond to the pre-calculated fingerprints which are stored in the first database.

Brief Description Of The Drawings

The invention will now be described in greater detail by reference to the drawings which show an example form of the invention. It is to be understood that the particularity of the drawings does not supersede the generality of the foregoing description of the invention.

Figure 1 is a schematic representation of a client portion and server portion of a security system on a Redhat Linux platform connected via a network according to a preferred embodiment of the present invention.

Figure 2 illustrates a more detailed data flow diagram relating to the schematic representation of Figure 1.

Detailed Description Of A Preferred Embodiment Of The Invention

Figure 1 is a schematic representation of a client portion and server portion of a security system on a Redhat Linux platform connected via a network 10 according to a preferred embodiment of the present invention. The system includes a client 12, a server 14 and a database of acceptable file fingerprints 16. Communication between the client 12 and server 14 may be via the Internet 18, using the TCP/IP protocol. The system is first set up by calculating and archiving fingerprints for all files relating to operating system or application software used in a typical Redhat Linux system, perhaps from original Redhat CDs or other secure software distribution methods. This software can be installed on test systems (not shown in Figure 1) so that the new files added or replaced can be fingerprinted and profiled. These new fingerprints, the file location of each file added or replaced, and other information, can then be stored in the database of acceptable file fingerprints 16. An alternative method that eradicates the need for software installation on a test system involves the use of a custom developed program that understands the RPM (Redhat Package Manager) format of software packages on the Redhat Linux CD. By examining each RPM software package's installation instructions, the program determines the file location and calculates the fingerprints of each file to be installed. This information and other information is then stored in the database of acceptable file fingerprints 16.

The fingerprints are preferably calculated using one or more cryptographic formulae. In the preferred embodiment, such cryptographic formulae may include hash functions to generate hash values for each file, or asymmetric cryptographic functions to generate digital signatures for each file. The original version of the files as well as patches, updates/upgrades of all types of operating system or application software should be fingerprinted. System performance and reliability will improve as more operating system and application software is fingerprinted and archived.

Hashing is a contraction of the file contents created by a cryptographic hash function. A hash value (or simply hash) is the output when an arbitrary input is passed into a hash function. The hash is substantially smaller than the input itself, and is generated by a formula in such a way that it is extremely unlikely that slight modifications of the input will result in the same hash. Hashes conventionally play a role in security systems where they are used to ensure that transmitted messages have not been tampered with. As an illustration, a sender generates a hash of the message and sends it with the message itself. The recipient then calculates another hash from the received message, and compares the two hashes. If they are the same, there is a very high probability that the message was transmitted intact. There may be other equivalent methods for calculating fingerprints that may be implemented as the relevant technology develops.

The system's client component is installed on the client 12 that requires file integrity protection. During the first time the client component is executed, the client software recurses through the file system and calculates and stores the cryptographic hash of every single file on the system. When the file system has been completely traversed, the client software makes a secure TCP/IP connection via the Internet 18 to the server component on the server, which usually resides on premises remote from the client component. However, the client component need not be physically located remote from the server component.

For security purposes, it is preferable that bi-directional authentication takes place between the client component and the server component before any further communication and this can be done with SSL (Secure Socket Layer) or TLS (Transport Layer Security). In a nutshell, the server presents its digital certificate to the client software and the client uses its hardwired CA (Certificate Authority) public credentials to verify the CA signature on the server's certificate. If the signature is authentic and the server's address matches the machine which the certificate was issued to, the client can be certain that the server is who it claims to be. Subsequently, the same thing happens in the reverse

direction. The client presents the server with its digital certificate and the server goes through the same process to verify that the client is who it claims to be. This practice is very common today and is an industry standard method of mutually authenticating two nodes communicating with one another. Other authentication methods may also be used.

The calculated hash results and gathered basic client system information from the client 12 are then transferred to the server 14 for validation. On the server 14, each hash result for each file on the client system is compared against what are the expected hash values given certain parameters such as the client system's operating system version and software patch/update level. This expected hash information is fetched from the database of acceptable file fingerprints 16 which houses all the pre-calculated hash values for all files in various operating systems and applications.

A report is then generated on the fly and returned to the client 12. This report lists the files on the client which are possibly unsafe since they do not represent authentic software from the vendor. There are 3 possible results for each file:

(a) the hash result matches so the given file on the client is definitely authentic;

(b) the database of acceptable fingerprints 16 has no information on such a file in the database and it is uncertain if the file is authentic;

(c) the hash result does not match the fingerprint in the database of acceptable fingerprints 16 and the file is suspicious.

Armed with this report, the systems administrator for the client server 12 can then verify each of the files in categories (b) and (c). Outcomes in categories (b) and (c) are typically from files that are part of an internal customer specific application that the database 16 will not contain. If the administrator verifies the hash with the owners of the application, the authenticity of the file can be determined. This should be done for all questionable files in the report so that a client system can be certified as 100% authentic. If some of the questionable files cannot be resolved via these means, it is likely that they have been

augmented by rogue software and should be replaced or the system should be reinstalled.

Using additional management software, the administrator can then check off all remaining questionable files as acceptable and the security system will take the additional hashes into account in all subsequent runs. These additional hashes can then be stored in a second database (not shown in Figure 1) so that they can be considered when checking other systems from the same customer - this is a configurable feature.

Figure 2 illustrates a more detailed data flow diagram relating to the schematic representation of Figure 1. Using the database of pre-calculated acceptable fingerprints 16, the system will be able to determine if any given file on a client's system is authentic, i.e. not invaded by rogue software. When comparisons are done, file location, time stamps, platform information, user preferences and other parameters can also be taken into consideration.

The system should be continuously updated with new fingerprint information in the database of acceptable file fingerprints 16 as new software and updates become available. The system thus provides pristine fingerprint information that is made available to the file integrity checking software installed on a client's computer system. Instead of identifying bad files, the system therefore ensures that the data is good. Instead of requiring users to have generated a fingerprint database some time back, the system provides pre-calculated fingerprints and greatly reduces the barriers to adoption of this important file integrity technology.

In addition to the above, the system may also store fingerprints of various customers' files in a separate database (not shown in Figure 1) so that the system can provide heuristic, statistically based best effort guesses on whether a certain fingerprint is acceptable for a given file.

In the above example, in the event of a questionable outcome (categories (b) and (c)) where the system does not have pre-calculated hash information on a

certain file on the client, the system may also render a heuristic result on whether a file is safe. This result can be provided by accessing the second database (not shown in Figure 1) which contains hashes that the customer's administrators have confirmed to be acceptable. For example, if the system does not know about whether a file such as "/usr/bin/myspecialprogram" should have a hash result of "xyz", it can inform the administrator, and also point out one of the following:

1) no other systems in the client's class have such a file so authenticity is uncertain;
2) other systems in the client's class which have such a file do not have a hash of "xyz" so the file is suspicious;
3) other systems in the client's class which have such a file have the same hash of "xyz" so the file is probably safe.

This information can also be provided with a percentage figure so that administrators have a best guess of where they stand before engaging in manual verification as described above.

As the client base grows, this information will allow the system to make increasingly improved guesses. The system can render an opinion along the lines of "10,000 other customers have this file and 9,985 of them have the same fingerprint, so your file is probably safe" - perhaps a common application whose fingerprint that does not already exist in the first database 16. Such information, while not substantive, allows users to zoom into more critical anomalies on their systems sooner. For example, consider this other response: "10,000 other customers have this file and no one has the same fingerprint as you. Worse yet, all these 10,000 customers have the same fingerprint so your file is most probably unsafe." The system can thus provide a percentage or quantifiable risk rating in either a numeric fashion or with the use of colours.

From the client's point of view, the advantage is that even systems currently deployed in risky public network environments can be easily reliably scanned and put onto a file integrity protection regime without re-installation to assure a pristine state and with significantly reduced downtime. As customers apply upgrades to their systems, the system will similarly be able to verify that new

software being installed is authentic since the fingerprints of the new software should be in the system's database 16. Conversely, the system can be programmed to warn the user if the update contains software the system does not believe is authentic.

While a particular embodiment of the invention has been shown and described, it will be obvious to those skilled in the art that changes and modifications of the present invention may be made without departing from the invention in its broader aspects. As such, the scope of the invention should not be limited by the particular embodiment and specific construction described herein but should be defined by the appended claims and equivalents thereof. Accordingly, the aim in the appended claims is to cover all such changes and modifications as fall within the spirit and scope of the invention.

Claims:

1. A method of detecting rogue software including the steps of:

(a) creating a first database containing pre-calculated fingerprints for each file relating to typical operating systems and application software, wherein the pre-calculated fingerprints are calculated using one or more cryptographic formulae;

(b) using the one or more cryptographic formulae to calculate fingerprints of files on a computer system which is to be scanned for rogue software;

(c) comparing the fingerprints calculated for the files on the computer system with the fingerprints which are contained in the first database of pre-calculated fingerprints;

(d) identifying files on the computer system which may contain rogue software by identifying files the calculated fingerprints of which do not correspond to the pre-calculated fingerprints which are stored in the first database.

2. A method according to claim 1 including the further step of generating a list of questionable files on the computer system for which the calculated fingerprints do not correspond to pre-calculated fingerprints stored in the first database.

3. A method according to claim 1, wherein the cryptographic formulae used in the pre-calculation of fingerprints which are stored in the first database and in the calculation of fingerprints for files on the computer system, use one or more hash functions to generate hash values for each file.

4. A method according to claim 1, wherein cryptographic formulae used in the pre-calculation of fingerprints which are stored in the first database and in the calculation of fingerprints for files on the computer system, use one or more asymmetric cryptographic functions to generate digital signatures for each file.

5. A method according to claim 2 wherein questionable files are considered by a system administrator and may be marked by the system administrator as acceptable.

6. A method according to claim 5 wherein the fingerprints for questionable files which are accepted by the system administrator are stored in a second database.

7. A method according to claim 6 which includes the step of calculating the probability that a questionable file has been corrupted by rogue software, by comparing its fingerprint with fingerprints for similar files that have previously been accepted by the system administrator and stored in the second database.

8. A method according to claim 7 wherein the system provides statistical information regarding:

(a) the number of fingerprints in the second database which represent files with the same characteristics as the questionable file;

(b) the number of fingerprints in the second database which are identical to the fingerprint of the questionable file;

(c) the number of fingerprints in the second database which are different to the fingerprint of the questionable file.

9. A method according to claim 5 wherein files that are not acceptable are replaced or reinstalled.

10. A method according to claim 6 wherein the step of calculating fingerprints for files on the computer system and the step of comparing fingerprints on the computer system with corresponding pre-calculated fingerprints stored on the first database are both implemented by the computer system and wherein verification of questionable files takes place before fingerprints from the computer system are added to the second database.

11. A method according to claim 10 wherein the computer system is physically remote from the first database and communication between the

computer system and the first database takes place over a network such as the Internet.

12. A method according to any one of claims 6 to 8 wherein the step of calculating fingerprints for the files on the computer system is implemented by the computer system and the step of comparing the fingerprints which represent files on the client system with corresponding pre-calculated fingerprints stored in the database is implemented by a server, and wherein verification of questionable files takes place between the computer system and the server before the corresponding fingerprints are transferred from the computer system to the second database.

13. A method according to claim 12 wherein the computer system is physically remote from the server and communication between them takes place over a communications network such as the Internet.

14. A system for detecting rogue software including:

(a) a first database containing pre-calculated fingerprints for each file relating to typical operating systems and application software, the fingerprints having been calculated using one or more cryptographic formulae;

(b) a software component which uses one or more cryptographic formulae to calculate fingerprints for files on a computer system; and

(c) a software component which compares the calculated fingerprints for the files on the computer system with corresponding pre-calculated fingerprints stored in the first database, such that files on the computer system which may contain rogue software are identified by identifying files the calculated fingerprints of which do not correspond to the pre-calculated fingerprints which are stored in the first database.

15. A system according to claim 14 including a software component which generates a list of questionable files for which the calculated fingerprints do not correspond to the pre-calculated fingerprints which are stored in the first database.

16. A system according to claim 14 or 15 wherein the software components are installed on the computer system.

17. A system according to claim 14 wherein the pre-calculation of fingerprints which are stored in the first database and calculation of fingerprints for files on the computer system use one or more hash functions to generate hash values for each file.

18. A system according to claim 14 wherein the pre-calculation of fingerprints which are stored in the database and calculation of fingerprints for files on the computer system use one or more asymmetric cryptographic functions to generate digital signatures for each file.

19. A system according to claim 15 further including a second database in which fingerprints of questionable files which are found to be acceptable by a system administrator are stored.

20. A system according to claim 15 wherein the system calculates the probability that a questionable file is a file that has been corrupted by rogue software, by comparing its fingerprint with fingerprints for similar files that have been verified and stored in the second database.

21. A system according to claim 20 wherein the system produces statistical information regarding:

(a) the number of fingerprints in the second database which represent files with the same characteristics as the questionable file;

(b) the number of fingerprints in the second database which are identical to the fingerprint of the questionable file;

(c) the number of fingerprints in the second database which are different to the fingerprint of the questionable file.

22. A system according to claim 19 wherein files that are not acceptable are replaced or reinstalled.

23. A system according to any one of claims 14 to 22 wherein the step of calculating fingerprints of files on the computer system and the step of comparing the fingerprints on the computer system with corresponding pre-calculated fingerprints stored on the database are both implemented by the computer system and wherein verification of questionable files takes place before fingerprints from the computer system are added to the second database.

24. A system according to claim 23 wherein the computer system is physically remote from the first database and communication between the computer system and the first database takes place over a network such as the Internet.

25. A system according to any one of claims 14 to 20 wherein the step of calculating fingerprints for the files on the computer system is implemented by the computer system and the step of comparing the fingerprints which represent files on the computer system with corresponding pre-calculated fingerprints stored in the first database is implemented by a server and wherein verification of questionable files takes place between the computer system and the server before the corresponding fingerprints are transferred between them.

26. A system according to claim 25 wherein the computer system is physically remote from the server and communication between them takes place over a communications network such as the Internet.

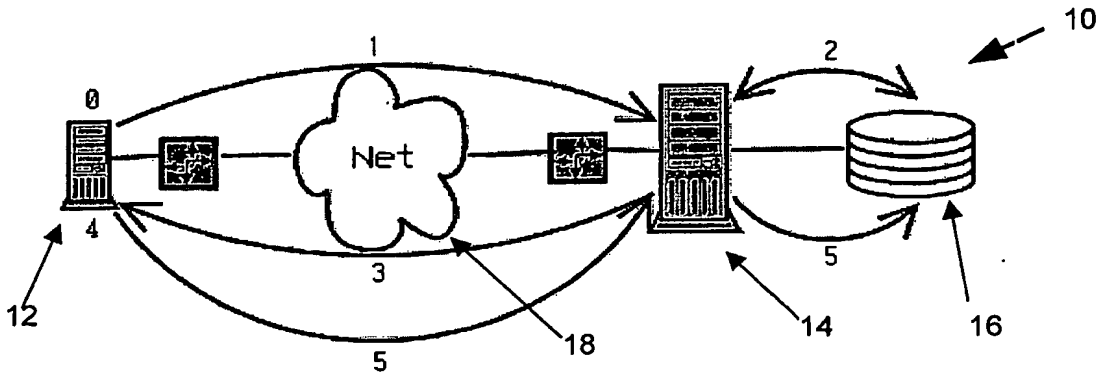


FIGURE 1

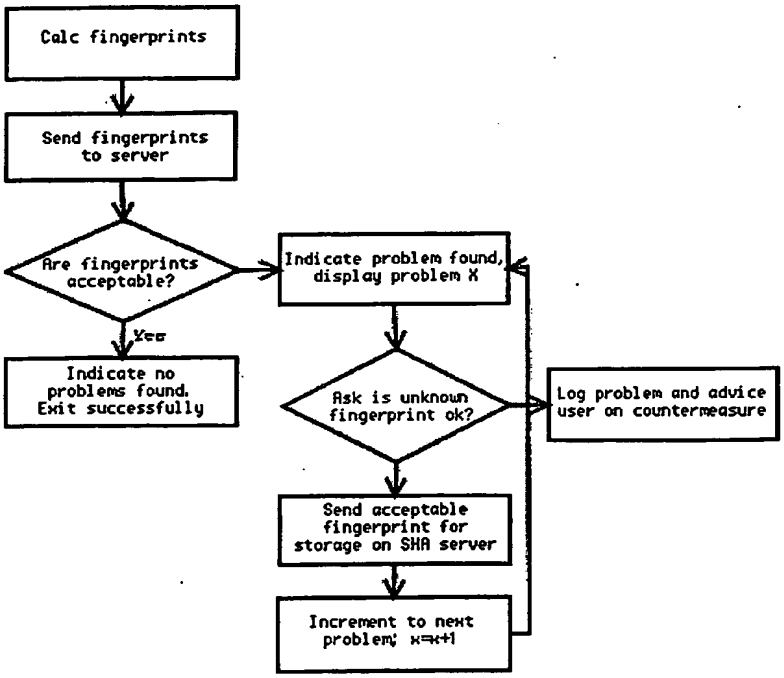


FIGURE 2

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☑ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
13 March 2003 (13.03.2003)

PCT

(10) International Publication Number
WO 03/021402 A2

(51) International Patent Classification[7]: G06F 1/00

(21) International Application Number: PCT/GB02/04059

(22) International Filing Date:
5 September 2002 (05.09.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
0121497.2 5 September 2001 (05.09.2001) GB

(71) Applicant *(for all designated States except US)*: **CRYPTIC SOFTWARE LIMITED** [GB/GB]; The Business Exchange, Rockingham Road, Kettering, Northants NN16 8JX (GB).

(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: **DUKE, David, John** [GB/GB]; 3 The Saplings, Kettering, Northants NN16 9UL (GB).

(74) Agent: **CRAWFORD, Andrew, B.**; AA Thornton & Co., 235 High Holborn, London WC1V 7LE (GB).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 03/021402 A2

(54) Title: NETWORK SECURITY

(57) Abstract: A computer network comprises a server computer and a plurality of local computers. Each local computer is provided with an engine program which monitors all files and peripherals and detects charges in the files or configuration. The results of the monitoring are used to create compressed information relating to the files and peripherals which is communicated to the server computer which stores previous information relating to the files and peripherals of the local computers. The server computer compares the received information with the stored information in order to detect the presence of altered and unwanted files or changes in configuration. In this way both known and unknown viruses can be detected as can un-authorised use of the local computers.

BEST AVAILABLE COPY

Trend Micro, Inc.
EX1007-684

NETWORK SECURITY

The present invention relates to an arrangement for monitoring the security of a computer especially computers on a network and particularly individual devices and information contained therein.

It is well known to provide individual computers with virus checking software but it is also well known that such software consumes resources when it is run and also takes time. Additionally, updating such software is somewhat cumbersome as each individual computer has to be updated separately.

Further, it is known to provide firewalls to protect computers and computer systems which are connected to a network or to the outside world but again these firewalls have limitations and they do not provide a completely effective defence to potential hackers.

Both of these methodologies are reactive and only detect threats to a network which have been seen and indexed previously.

It is an object of the present invention to provide an arrangement whereby protection from attack by viruses is improved as well as providing a more effective firewall in the event that the computer or computer system is provided with such. It is a further object that the invention is deployed in such a way that it is itself not vulnerable to attack.

From one aspect, the present invention is based on a concept whereby a main memory holds a record of all files used by a local computer, is provided with update information on all existing files as well as all new files and determines whether the updated files or new files represent a security threat.

In a preferred embodiment, as data is created, modified or deleted on the local computer, events are monitored and collected by an engine program and forwarded to the main memory. Preferably, the engine program itself consists of a

single file which is arranged to run a program in parallel with the normal operating system of the local computer thus making the file easier to hide so that the presence of the engine program will not be noticed either by the user or by a hacker. Further, as far as the user of the local computer is concerned, the engine program has no noticeable effect on the system. Preferably, the main memory is provided by a central computer (hereafter referred to as the console) and the local computer communicates with the console over a network or in some other convenient manner.

The engine program monitors each and every file and by inspecting preselected parts of each file can create a highly compressed accurate image of the file which can then be transmitted to the central database. It will be appreciated that all files can be handled in this way. Consequently, when the term "file" is used we mean all files associated with the operating system, data, registries, directories, hardware, software and such like. In this way, a complete virtual image of a local computer can be stored within the central database and the console can have management software for monitoring changes in the data files, programs or hardware of the local computer and thus warn of tampering with such files, programs and/or hardware.

From another aspect, the present invention provides a method of monitoring the security of a computer system comprising monitoring a file as it is created or updated by inspecting preselected portions of the file selected from the whole file, storing information derived from the preselected portions and transmitting the stored information to a main memory location. The main memory location is preferably a central database of a network.

From a further aspect the present invention provides a method of improving the security of computer apparatus by providing two copies of a security program and interconnecting them such that as one is switched off, either deliberately or

inadvertently, the other is automatically switched on and vice versa. It is to be understood that this aspect is not limited to any particular type of security program and is of general application. However, it has particular application with the present invention where the security program is relatively small and easy to hide.

In order that the present invention be more readily understood, an embodiment thereof will now be described with reference to the accompanying drawings in which: -

Fig 1 shows a block diagram of a network according to the present invention;

Fig 2 shows a block diagram representing the main functions of an engine program arranged to be installed on each computer to be protected; and

Fig 3 is a flow chart for explaining the operation of the engine program of Fig 1.

The present invention will be described in relation to its use for network security but it will be appreciated that the invention can be implemented in a number of different ways independently of any connection technology. In fact, the totality of the invention could be located within a single computer or a local computer could be in communication with a central server over conventional telephone lines rather than utilising any network protocol.

As shown in Fig 1, a central server 1 communicates via transmit/receive ports 2 with a number of local computers 3, each of which is provided with a monitoring engine program 4. The central server is provided with a plurality of databases, namely an archive of threats database 5, a virtual image of each computer database 6 and an audit database 7.

The basis of the present embodiment is that a virtual copy of a local computer and all its files will be kept in memory at the database to the central server computer with each file having its own individual characteristics stored at

the central computer. The virtual copies are created by the small engine programs 4 loaded on the local computers 3 which monitor all files by inspecting preselected portions of the files in order to create a so-called "fingerprint" of each file which is in fact a highly compressed version of the file. The fingerprint can then be very speedily forwarded to the central location where the current fingerprint can be compared with a previous fingerprint and any changes detected. The changes are then evaluated by the central computer 1 in order to determine the level of threat, if any, to the security of the local computer system and an appropriate signal sent back from the central server to the local computer 3 in the event that a particular file should not be opened or a particular program should not be run. Simultaneously, an event is displayed at the central computer and/or forwarded to other programs.

It will be appreciated that by the use of a virtual copy of the local computer and all its files, the central server can then monitor and log all changes to files, programs and/or hardware in order to provide evidence of breaches of security at the local computer. As an example of this, one could consider a computer fitted with several pieces of memory. Removal of all or part of this memory is an event which can be detected by the engine program and thus signalled to the central computer which can log the event.

In further explanation of the evidential facilities of this invention on the triggering of specified events, biometric information can be gathered in real time and forwarded to the central database. For example, if an event is detected, a screen shot of the local computer is gathered for evidence and at the same time the engine program can gather biometric information such as a photograph of the user of the local computer.

The above description represents a simple brief overall view of the arrangement in accordance with the present invention but for a more detailed

description we will now separate the operation of the engine program at the local computer and firstly describe this before we move onto a description of the operation of the monitoring program at the central server.

The engine program is shown diagrammatically in Fig 2 where all file data is monitored at monitoring section 11 under the control of a program section 12 which determines which parts of the data will be sampled. The selected parts of the data are then stored as indicated at 14 in order to create a virtual compressed copy of the file data for transmission as indicated at 16.

The engine program is arranged to commence running as soon as the computer on which it is loaded is powered up. The program then monitors all files as they are opened and in particular it monitors all changes to a file as indicated by the flow chart shown in Fig 3. Each and every change is inspected in order to determine whether the change is the creation of a new file. It also determines whether the change is a valid change. Each file is also checked by inspecting the file from a plurality of points of view by sampling predetermined portions of the file. This in turn creates a fingerprint or virtual copy of the file which is then assembled for transmission to a central database either over the network or via some other communications link. Among the predetermined portions of the file which the engine program inspects is the tag indicating whether or not the file is an executable. It also determines whether the file is a manipulated file, eg a zipped or encrypted file. It looks for the presence of capabilities such as keystroke logging, FTP server capability, IP notification, joystick controls, game libraries, etc

The engine program has a memory capability 12 so that it can store instructions received from the central processor as described below and then carry out those instructions in the event that the fingerprint it has assembled of a file indicates that action is necessary. As an example, if a known threat is present in a

file as indicated by the fingerprint, the engine program can immediately kill the threat in view of the fact that it has the necessary instruction previously received from the central server. In certain circumstances, the file in question can be automatically dealt with.

A particular feature of the engine program is that it does not wait to be polled by a central computer. Rather, it itself generates a message for transmission to the central computer. This has the advantage that the engine program is immune to attack from a hacker because there is no "listening" port waiting for an incoming transmission. It is not until the engine program has communicated with the central server that a two way communication is possible and only after the engine program's communication with the central server can the central server transmit modified information to the engine program. However, the monitoring and modification of the files on the local computer occur in real time while the local computer is in operation either by being started up or while the local computer is running programs. Further, the engine program can itself run other software in order to transmit and/or record data as a result of the detection of a particular event or change to a file. As an example of this, if the local computer is fitted with a web cam, should particular files be modified, the engine program can take a picture of the user of the machine at the appropriate time and correlate the event with the actual user as evidence for subsequent use.

In an especially secure system, it is possible to provide two identical engine programs which are interlinked with each other so that in the event that one is disabled for any reason, eg being switched off, the other is activated before the first is terminated and vice versa. This ensures security in the event that one of the engine programs is detected by a hacker or other person seeking to tamper with the local computer.

Turning now to the central server 1, this contains a reception port 2 for receiving data transmitted from the local computer. As fingerprints are stored, the central computer builds up a virtual picture of each local computer, its hardware, programs and files generally in its database 6. In addition, the central server maintains a database 5 of all known security threats and viruses. As fingerprints of files are received these are compared with the information in the threats database 5 so that any threat can be detected instantaneously. Additionally, because the central server maintains an up to date virtual copy of each local computer with which it is in communication, in the event that it identifies a security threat in relation to one computer 3, it can then inspect all other local computers for similar configurations and then be ready to communicate the necessary instruction to another local computer 3 as soon as the other local computer communicates with the central server for any reason. In this way, security threats and the mechanism of deleting files or repairing infected files can be undertaken much more quickly and in fact even before a local computer might have been subjected to attack or infection. In other words, when a new threat is identified, it is added to the server database so that the central processor can check its data relating to the local computers to which it is connected ot identify other infected systems without the need to communicate with them. Therefore even if a laptop is disconnected or machines are powered down, threats can still be identified and remedial action taken once such machines are back in communication with the central server.

In the same way, the central computer can monitor changes to hardware and software using the information in the database 7 and consequently know when a security breach might have taken place. In this case, the central computer can log the event in question or signal the local computer to take the necessary action to

provide evidence of the possible security breach, eg by taking a web cam picture as well as recording the user name and password.

Another sophisticated ability of the server is to construct a dynamic accurate focused detection formula which can be custom built to detect dynamic changing threats which avoid typical security fingerprinting techniques.

The central server can communicate back to the local computer in the event of a security breach or potential security breach or it can communicate with some other communications device such as a mobile telephone or personal digital assistant by sending an e-mail SMS or fax to indicate the existence of the security breach or potential security breach. In this way, users of portable computers could be warned of security breaches relating to their machine without the need to actually switch it on and establish communications with the central server.

CLAIMS:

1. A computer arrangement comprising a central processor, first memory means for storing an operating system program for the central processor, further memory means for storing data relating to all files used by the computer arrangement, storage means separate from the first storage memory means for storing an engine program for monitoring files used or created by the computer arrangement, and means for communicating the results of the monitoring to the further memory means.

2. A computer arrangement according to claim 1, wherein the engine program is arranged to inspect preselected parts of each file and create a compressed version of the file.

3. A computer arrangement according to claim 1, wherein the further memory means is located remote from the central processor and first memory means.

4. A computer arrangement according to claim 3, wherein the means for communicating results to the further memory means comprises a network connection.

5. A computer arrangement according to claim 1, wherein the storage means is arranged to store two engine programs each of which is capable of being switched off and which are interconnected such that as one is switched off, the other is switched on.

6. A computer arrangement according to claim 1, wherein the engine program is arranged to monitor peripheral devices connected to the arrangement in order to detect the presence of new devices and the absence of previously connected devices.

7. A computer arrangement according to claim 6, wherein the engine program is arranged to record data relating to any change in peripheral devices or files.

8. A computer arrangement according to claim 3, wherein the further memory means is located in a further computer which is arranged to analyse the results of the monitoring.

9. A computer arrangement according to claim 8, wherein the further computer is arranged to transmit instructions to the central processor depending on the results of the analysis.

1/2

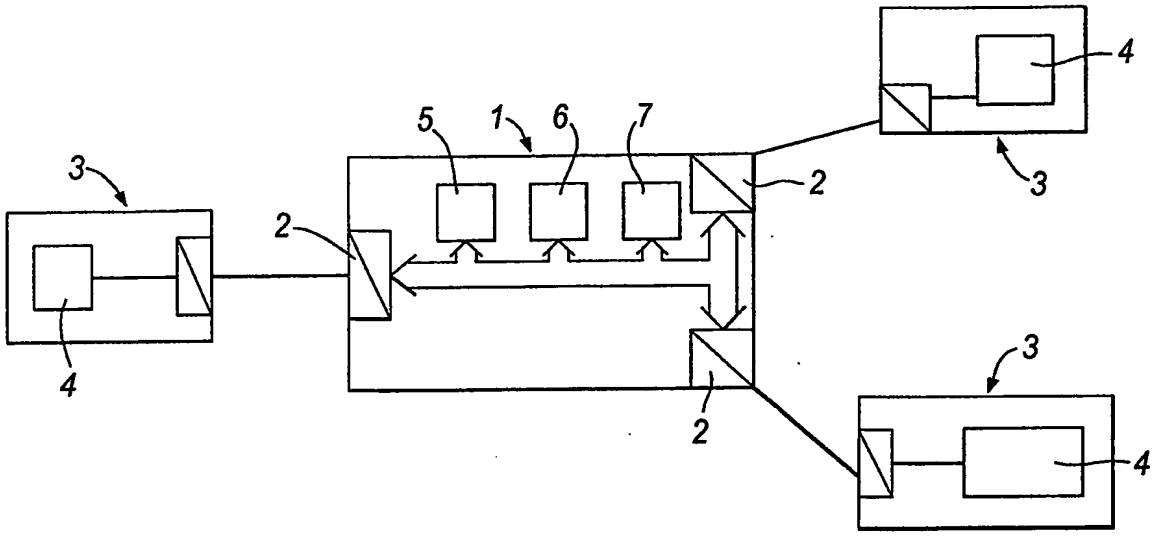


Fig.1

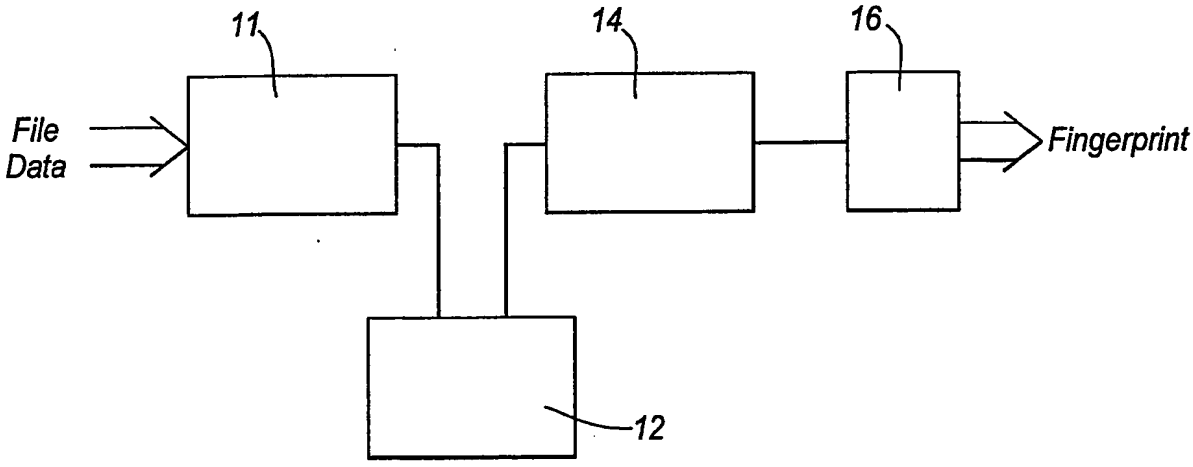


Fig.2

Trend Micro, Inc.
EX1007-695

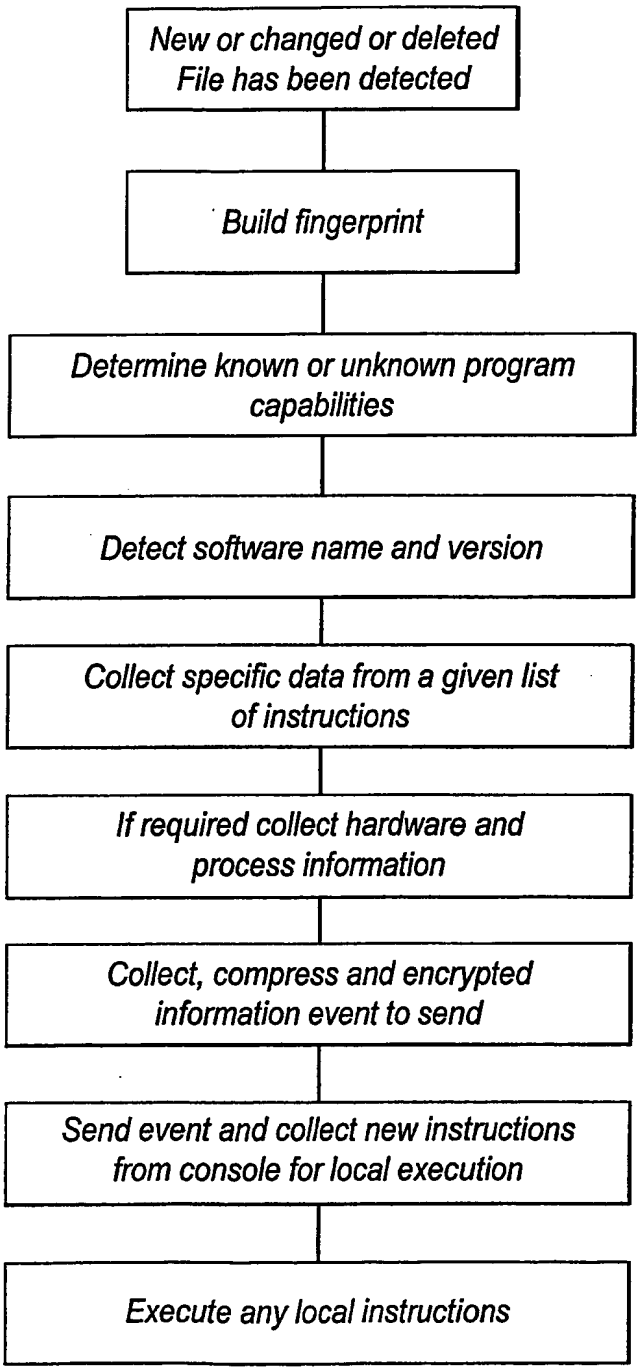


Fig.3

Trend Micro, Inc.
EX1007-696

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☑ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 280 040 A3

(12) EUROPEAN PATENT APPLICATION

(88) Date of publication A3:
24.03.2004 Bulletin 2004/13

(43) Date of publication A2:
29.01.2003 Bulletin 2003/05

(21) Application number: **02253235.2**

(22) Date of filing: **09.05.2002**

(51) Int Cl.[7]: **G06F 1/00**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **26.07.2001 US 912392**

(71) Applicant: **Networks Associates Technology, Inc.**
Santa Clara, CA 95054 (US)

(72) Inventors:
- **Hinchliffe, Alexander James**
  Milton Keynes, Buckinghamshire MK7 7LL (GB)
- **Howard, Fraser Peter**
  Chipping Norton, Oxon OX7 3BT (GB)
- **Kemp, Andrew**
  Oxford, Oxon OX4 2QT (GB)
- **Rai, Bobby**
  Kettering, Northants NN15 5BP (GB)

(74) Representative:
Robinson, Nigel Alexander Julian
D. Young & Co.,
21 New Fetter Lane
London EC4A 1DA (GB)

(54) Centrally managed virus scanning system

(57) A malware detecting system is described in which a client computer 8 sends a file access clearance request to an assessment computer 24 to determine whether or not access is permitted to that file. In this way, the different client computers on a network 2 may share their scan results and centralised control may be provided over access permissions to individual files or groups of files.

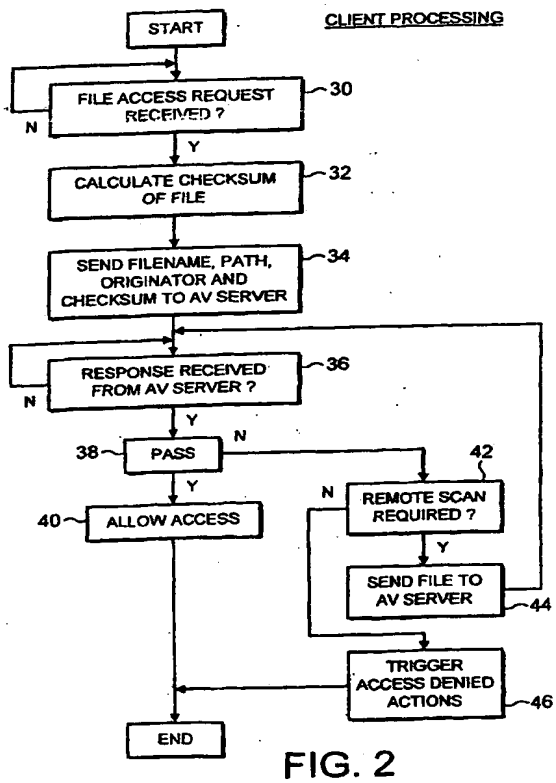


FIG. 2

BEST AVAILABLE COPY

EP 1 280 040 A3

Trend Micro, Inc.
EX1007-698

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 02 25 3235

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
X	US 6 195 587 B1 (HRUSKA JAN ET AL) 27 February 2001 (2001-02-27) * abstract * * column 1, line 1-9; figure 3 * * column 2, line 15-65; figures 1,2 * * column 3, line 5-25,40-67 * * column 5, line 1-60 * * column 4 *	1-13, 16-24, 28-41, 44-52, 55-68, 71-79,82	G06F1/00 G06F21/00
X	US 5 960 170 A (LAU STEVEN YUEN-LAM ET AL) 28 September 1999 (1999-09-28) * abstract * * column 1-10; figures 1-8 * * column 11, line 1-55 * * column 12-27 *	1,7,17, 29,35, 45,56, 62,72	TECHNICAL FIELDS SEARCHED (Int.Cl.7)
X	EP 1 111 507 A (SECURE OYJ F) 27 June 2001 (2001-06-27) * the whole document *	1,7,17, 29,35, 45,56, 62,72	G06F

-/--

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
MUNICH	26 January 2004	Nazzaro, A

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

Trend Micro, Inc.
EX1007-699

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 02 25 3235

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
A	MARMELSTEIN R E ET AL: "A distributed architecture for an adaptive computer virus immune system" SYSTEMS, MAN, AND CYBERNETICS, 1998.. 1998. IEEE INTERNATIONAL CONFERENCE ON SAN DIEGO, CA, USA 11-14 OCT. 1998, NEW YORK, NY, USA,IEEE, US, 11 October 1998 (1998-10-11), pages 3838-3843, XP010310923 ISBN: 0-7803-4778-1 * abstract; figure 1 * * page 3841 * * page 3842 *	14,15, 25,26, 42,43, 53,54, 69,70, 80,81	

TECHNICAL FIELDS SEARCHED (Int.Cl.7)

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
MUNICH	26 January 2004	Nazzaro, A

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

Trend Micro, Inc.
EX1007-700

European Patent Office

Application Number
EP 02 25 3235

CLAIMS INCURRING FEES

The present European patent application comprised at the time of filing more than ten claims.

☐ Only part of the claims have been paid within the prescribed time limit. The present European search report has been drawn up for the first ten claims and for those claims for which claims fees have been paid, namely claim(s):

☐ No claims fees have been paid within the prescribed time limit. The present European search report has been drawn up for the first ten claims.

LACK OF UNITY OF INVENTION

The Search Division considers that the present European patent application does not comply with the requirements of unity of invention and relates to several inventions or groups of inventions, namely:

see sheet B

☒ All further search fees have been paid within the fixed time limit. The present European search report has been drawn up for all claims.

☐ As all searchable claims could be searched without effort justifying an additional fee, the Search Division did not invite payment of any additional fee.

☐ Only part of the further search fees have been paid within the fixed time limit. The present European search report has been drawn up for those parts of the European patent application which relate to the inventions in respect of which search fees have been paid, namely claims:

☐ None of the further search fees have been paid within the fixed time limit. The present European search report has been drawn up for those parts of the European patent application which relate to the invention first mentioned in the claims, namely claims:

Trend Micro, Inc.
EX1007-701

European Patent Office

LACK OF UNITY OF INVENTION
SHEET B

Application Number
EP 02 25 3235

The Search Division considers that the present European patent application does not comply with the requirements of unity of invention and relates to several inventions or groups of inventions, namely:

1. Claims: 1-13,16-24,28-41,44-52,55-68,71-79,82

Computer program , method and apparatus for detecting malware, comprising: detecting logic, file access clearance request generating and transmitting logic, file access clearance receiving logic at an assessment computer, file access clearance response generating and transmitting logic, file access clearance response receiving logic and file access permitting logic if file access clearance response indicates said file does not contain malware.

2. Claims: 14-15, 25-26, 42-43, 53-54, 69-70, 80-81

Computer program method and apparatus for detecting malware comprising detecting logic, file access clearance request generating and transmitting logic, file access clearance receiving logic at an assessment computer, file access clearance response generating and transmitting logic, file access clearance response receiving logic and file access permitting logic if file access clearance response indicates said file does not contain malware. The assessment computer is operable in a higher level security mode serving to deny access to greater range of computer files. The change from lower to higher level security mode is triggered from a remote computer.

Trend Micro, Inc.
EX1007-702

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 02 25 3235

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

26-01-2004

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 6195587 B1	27-02-2001	GB 2283341 A	03-05-1995
		AU 8000394 A	22-05-1995
		DE 69407812 D1	12-02-1998
		DE 69407812 T2	16-07-1998
		EP 0725951 A1	14-08-1996
		WO 9512162 A1	04-05-1995
		JP 9504395 T	28-04-1997
US 5960170 A	28-09-1999	AU 6537698 A	12-10-1998
		EP 0968466 A1	05-01-2000
		JP 2001515625 T	18-09-2001
		WO 9841919 A1	24-09-1998
EP 1111507 A	27-06-2001	GB 2353372 A	21-02-2001
		EP 1111507 A2	27-06-2001
		US 2001005889 A1	28-06-2001

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

Trend Micro, Inc.
EX1007-703

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 280 040 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
29.01.2003 Bulletin 2003/05

(51) Int Cl.[7]: **G06F 1/00**

(21) Application number: **02253235.2**

(22) Date of filing: **09.05.2002**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **26.07.2001 US 912392**

(71) Applicant: **Networks Associates Technology Inc.**
Santa Clara, CA 95054 (US)

(72) Inventors:
- **Hinchliffe, Alexander James**
  Milton Keynes, Buckinghamshire MK7 7LL (GB)
- **Howard, Fraser Peter**
  Chipping Norton, Oxon OX7 3BT (GB)
- **Kemp, Andrew**
  Oxford, Oxon OX4 2QT (GB)
- **Rai, Bobby**
  Kettering, Northants NN15 5BP (GB)

(74) Representative:
Robinson, Nigel Alexander Julian
D. Young & Co.,
21 New Fetter Lane
London EC4A 1DA (GB)

(54) Centrally managed virus scanning system

(57) A malware detecting system is described in which a client computer 8 sends a file access clearance request to an assessment computer 24 to determine whether or not access is permitted to that file. In this way, the different client computers on a network 2 may share their scan results and centralised control may be provided over access permissions to individual files or groups of files.

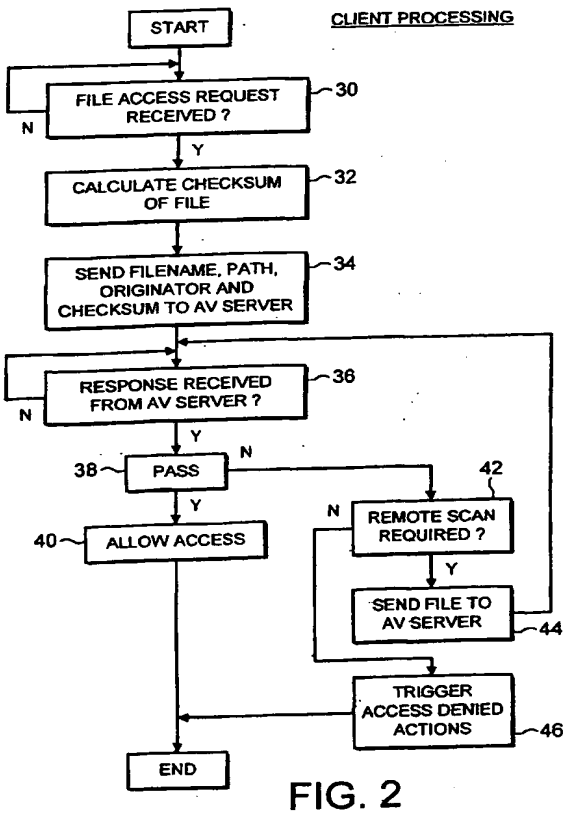


FIG. 2

EP 1 280 040 A2

Trend Micro, Inc.
EX1007-704

Description

[0001] This invention relates to the field of data processing systems. More particularly, this invention relates to the field of malware scanning, such as, for example, scanning for computer viruses, Trojans, worms, banned computer files or E-mails containing banned words or content.

[0002] It is known to provide malware scanning systems and mechanisms for identifying malware within a computer file to be accessed. As the number of computer viruses and the like that are present in the wild increases the processing associated with scanning a computer file to identify the presence of any of those viruses similarly increases. This increase in the processing required is disadvantageous. Furthermore, with the increasing levels of network connectivity between computer systems and the use of E-mail and other fast messaging systems, the spread of computer viruses has become increasingly rapid. In a number of recent cases mass mailing viruses have spread at such a speed that considerable damage has been caused before appropriate countermeasures have been able to be put in place. The delay in deploying such countermeasures is further increased by the need in many cases to update virus definition data on every individual node computer to be protected. The speed of spread of recent viruses is such that the download and installation delays associated with installing such new countermeasures, even when they have been developed and are available, is a significant disadvantage.

[0003] An anti-virus system produced by Sophos uses a locally held record of previously conducted on-access scans on the individual computer in question to determine whether or not a scan should be conducted for a computer file or a previously determined result used instead when this is available.

[0004] Viewed from one aspect the present invention provides a computer program product for controlling a computer to detect malware, said computer program product comprising:

detecting logic operable to detect a file access request to a computer file by a requesting computer;
file access clearance request generating logic operable to generate a file access clearance request including data identifying said computer file;
file access clearance request transmitting logic operable to transmit said file access clearance request from said requesting computer to an assessment computer responsible for assessment of whether said computer file contains malware;
file access clearance response receiving logic operable to receive at said requesting computer a file access clearance response from said assessment computer; and
file access permitting logic operable if said file access clearance response indicates said computer file does not contain malware, to permit said file access request by said requesting computer.

[0005] The invention recognises that an individual computer requiring a file access may pass the task of determining whether or not that access should be allowed to a different assessment computer. Whilst at first sight this may seem that it would slow down the file access, in practice there are considerable advantages. For example, the assessment computer to which the task is passed may be more rapidly updated with new virus definition data as this is released than would be possible for the requesting computer. Accordingly, the requesting computer can benefit from the most up-to-date virus definition data more rapidly. Furthermore, the invention is particularly well suited to systems in which a plurality of requesting computers share an assessment computer since in many cases the individual requesting computers will show a high degree of correlation in the computer files to which they are requesting access and for which a malware scan is needed. Thus, rather than each requesting computer individually scanning the computer files that are also being scanned by a large number of other computers, (e.g. the computer files associated with the operating system shared by all the requesting computers), these computer files can instead be scanned once by the assessment computer and then each requesting computer can check that the computer file has been scanned and cleared by sending an appropriate request to the assessment computer rather than having to scan the file itself. Thus, in exchange for the use of a small amount of bandwidth on the connections between the requesting computers and the assessment computer, a considerable processing burden may be lifted from the requesting computers. The additional processing burden on the assessment computer does not increase disproportionately since the high degree of correlation between the computer files accessed by the different requesting computers means that in many cases the assessment computer will be able to respond to a clearance request from a requesting computer on the basis that it has already scanned that file and without the need to rescan that file.

[0006] In order to identify the computer file to be scanned in a secure manner to the assessment computer, the data identifying the computer file preferably includes a checksum value. Checksum values may be made sufficiently specific for a computer file from which they have been calculated so as to be difficult to bypass as a way of uniquely identifying a computer file.

[0007] Additional data that is highly useful to the assessment computer in managing the requests it receives includes the filename of the computer file, data identifying the requesting computer and the storage location of the computer file. This provides useful audit information as well as providing for the possibility of files having the same name and same storage location on different computers in fact being different files with dif-

Trend Micro, Inc.
EX1007-705

ferent checksums, both of which should be treated as separate entities when determining whether or not a request from another requesting computer relates to a file that has already scanned.

[0008] In circumstances when a requesting computer wishes to access a file that has not already been scanned, the assessment computer may send a scan request message back to the requesting computer such that the requesting computer may send a copy of the file to the assessment computer for scanning. It will be appreciated that this may be slower than scanning the file locally, but the benefit will be that should a different requesting computer later wish to access that same file, then the scanning result produced by this action can be shared with that other computer, so speeding the operation of that other computer.

[0009] If the assessment computer determines that access should be denied to a computer file, then this may be used to trigger denied access actions within the requesting computer, the assessment computer, or elsewhere. Such denied access actions may include deletion of the computer file, repair of the computer file, quarantining of the computer file, generation of user warning messages, generation of administrator warning messages and the like.

[0010] Viewed from another aspect the invention also provides a computer program product for controlling a computer to detect malware, said computer program product comprising:

file access request receiving logic operable to receive at an assessment computer a file access clearance request from a requesting computer, said file access clearance request including data identifying a computer file to be accessed by said requesting computer;

file access clearance response generating logic operable in dependence upon said data identifying said computer file to determine if said computer file has previously been assessed as not containing malware and to generate a file access clearance response; and

file access clearance response transmitting logic operable to transmit said file access clearance response to said requesting computer.

[0011] The assessment computer may advantageously store the results of its previous scans within a database. This database may include a variety of fields relating to each computer file that has been scanned. These fields may include an access flag indicating whether access is to be denied to that computer file and a persistence flag indicating whether or not the entry for that computer file should be purged during purge operations.

[0012] The access flag is particularly useful as in addition to allowing the recording that access should be denied due to malware being detected, it also allows the central management of whether or not a particular individual file or class or type of file should be permitted to be accessed by all of those requesting computers that seek their access permissions from that assessment computer. This centralised control is a powerful tool that may be used to implement techniques such as the triggering of a lock down mode of operation in which higher level security provisions are put in place by denying access to certain files or types of files. As an example, if a message was received indicating that the higher security mode should be entered, then the assessment computer may use its mechanisms to deny access to any newly encountered computer file that had not previously been scanned and cleared for use. This would typically allow the large majority of computer activity to continue whilst providing protection against newly released malware threats until the appropriate countermeasures could be put in place.

[0013] The persistence flag allows control of the flushing of entries from the assessment computer. Whilst one of the advantages of the invention is storing the results of previously conducted scans such that they need not be repeated, this has to be tempered by allowing the results to be refreshed at a later time for at least some files. It is possible to envisage that a particular computer file carrying a newly released virus may not be detected as carrying that virus when it is first scanned, but later when a new virus driver is available, that computer file would be detected and blocked. Accordingly, as an example, it may be that all previous scan results could be purged from the system whenever the virus definition data was updated.

[0014] Viewed from a further aspect the invention provides a computer program product for controlling a computer to detect malware, said computer program product comprising:

file access request detecting logic operable to detect a file access request to a computer file by a requesting computer;

file access clearance request generating logic operable to generate a file access clearance request including data identifying said computer file;

file access clearance request transmitting logic operable to transmit said file access clearance request from said requesting computer to an assessment computer responsible for assessment of whether said computer file contains malware;

file access clearance request receiving logic operable to receive at said assessment computer said file access clearance request from a requesting computer;

file access clearance response generating logic operable in dependence upon said data identifying said computer file to determine if said computer file has previously been assessed as not containing malware and to generate a file access clearance response;

Trend Micro, Inc.
EX1007-706

file access clearance response transmitting logic operable to transmit said file access clearance response to said requesting computer;
file access clearance response receiving logic operable to receive at said requesting computer said file access clearance response from said assessment computer; and
file access permitting logic operable if said file access clearance response indicates said computer file does not contain malware to permit said file access request by said requesting computer.

[0015] As well as the complementary aspects of the invention embodied in the form of the client software, the server software and the combination of the client and server software, the invention may also take the form of corresponding methods of malware detection and apparatus for malware detection.

[0016] Embodiments of the invention will now be described, by way of example only, with reference to the accompanying drawings in which:

Figure 1 schematically illustrates a computer network to which the present technique may be applied;

Figure 2 is a flow diagram illustrating an example of the processing that may be performed on a client computer;

Figure 3 is a flow diagram illustrating an example of the processing that may be performed on a server computer;

Figure 4 is a flow diagram illustrating an example process that may be run on a server computer waiting for a lock down signal;

Figure 5 schematically illustrates a database entry relating to a computer file that has previously been scanned; and

Figure 6 is a diagram schematically illustrating the form of a general purpose computer that may be used to implement the above described techniques.

[0017] Figure 1 schematically illustrates a computer network 2 comprising a plurality of client computers 4, 6, 8, 10, 12, 14, 16, 18 connected to respective departmental servers 20, 22 and to a corporate anti-virus (assessment) server 24. The computer network 2 is also connected via a network link to an anti-virus providers FTP server 26 from which virus definition data may be downloaded and from which messages, such as lock down messages or messages relating to the availability of new virus definition data may be received.

[0018] In operation a client computer 8 conducts processing activity during which it seeks to make a file access to a particular computer file (in this example, winword.exe). Rather than scanning this file itself for malware such as computer viruses, Trojans, worms, banned content etc, the client computer 8 instead passes an access clearance request relating to the computer file through to the assessment computer 24 via the departmental server 20. The access clearance request includes the file name of the computer file being accessed, a checksum derived from that file in an effort to uniquely identify it (e.g. an MD5 checksum), data identifying the name of the client computer 8, and the path used by that client computer 8 to access the computer file. The assessment computer 24 receives the access clearance request and uses the data identifying the computer file to look up within an associated database 28 as to whether or not a malware scan has already been conducted for that particular computer file. The computer file is uniquely identified by its filename and checksum value. If the computer file in question has already been scanned, then the result of that scan may be reused rather than conducting the scan again. The assessment computer 24 can accordingly pass back to the client computer 8 a clearance request response indicating whether or not access to that file is permitted or some other further action should be taken. The client computer 8 can then use this access clearance response to either permit access to the file or take the further action specified.

[0019] The further action specified could be that the computer file has already been identified as containing malware and accordingly an appropriate anti-virus or other anti-malware response should be initiated on the client computer 8. A corresponding response could already have been initiated on the assessment computer 24. The denied access responses could take the form of deleting the computer file concerned, repairing the computer file concerned, quarantining the computer file concerned, issuing a user alerting message, issuing an administrator alerting message, issuing a message to an anti-virus provider or some other action.

[0020] A different type of further action that could be required by the client computer 8 in response to the access clearance response arises when the assessment computer 24 has not previously scanned that file and accordingly the client computer 8 should send a copy of that file through to the assessment computer 24 to be scanned. When such a copy of the computer file has been sent through to the assessment computer 24 and scanned, an appropriate pass or denied access action triggering response can be sent back from the assessment computer 24 to the client computer 8.

[0021] It will be appreciated that the client computers 4, 6, 8, 10, 12, 14, 16, 18 operate to share the results of the malware scanning that is performed on their behalf by the assessment computer 24. Since there is likely to be a high degree of correlation between the files being accessed by the different client computers, in many cases the scan that a client computer is request-

Trend Micro, Inc.
EX1007-707

ing will already have been performed on behalf of another computer and that scan need not be run again by the assessment computer but instead merely the result of the previous scan returned. The use of a checksum allows the computer files to be reliably uniquely identified.

[0022] The co-ordination of file access permissions by the assessment computer 24 also allows central management of which computer files may be accessed upon the network 2 in a manner that allows the rapid implementation of any higher security level mode should this be desired. As an example, a lock down trigger message may be received from the anti-virus provider FTP server 26 that will trigger the assessment computer 24 to enter a higher security level mode compared to its normal mode of operation. In the higher security level mode, it may be that the access to whole variety of different types of files may be temporarily banned. As an example, the access to VBS files which are often a source of viruses may be banned across the network 2, as could access to e-mail attachments which are another potential source of virus propagation. Thus, when a new threat is identified by the anti-virus provider, lock down messages may be used to trigger predetermined, in accordance with user configuration and preferences, higher security level modes within connected assessment computers 24 to provide a degree of protection for the networks concerned whilst allowing the majority of their normal operations to continue.

[0023] Figure 2 is a flow diagram illustrating the processing performed by a client computer. At step 30 the computer waits to receive a file access request from a computer program executing on that client computer. At step 32 a checksum value is calculated in accordance with one of several different possible checksum algorithms, such as the MD5 algorithm. At step 34 the filename, path, originating computer name and checksum value are sent through to the assessment computer. At step 36 the client computer waits to receive a response from the assessment computer. When that response is received, step 38 serves to determine whether or not the response indicates that the computer file had passed its scanning. If the computer file had passed its scanning and access is to be permitted, then processing proceeds to step 40 at which the access is allowed.

[0024] If the response is not that the computer file had passed its scan, then processing proceeds to step 42 at which a test is made as to whether or not the response indicated that a remote scan was required. If a remote scan was required, then the computer file in question is sent to the assessment computer at step 44 and processing is returned to step 36.

[0025] If the test at step 42 does not identify a remote scan request, then since the response is neither a pass, or a remote scan request, then the computer file must be one to which access is denied and accordingly step 46 serves to trigger the denied access actions. These may include deletion, repair, quarantining or other actions upon the computer file in question as well as the generation of appropriate warning messages to a user or an administrator.

[0026] Figure 3 illustrates the processing performed by the assessment computer. At step 48 the assessment computer waits to receive a request from a client computer. When a request is received, step 50 serves to identify whether a copy of a computer file is being returned to the assessment computer for scanning following a remote scan request that had earlier been issued by the assessment computer. If a copy of a computer file is being returned for scanning, then processing proceeds to step 52 at which the necessary malware scanning is conducted. Step 54 determines whether or not the computer file passed this malware scanning. If the computer file did pass the malware scanning, then step 56 serves to add details of that computer file to the database of scanned files held at the assessment computer relating to scans already performed. Step 58 then returns the pass result to the client computer.

[0027] If the scanning at step 52 was detected as not being passed at step 54, then processing proceeds to step 60 at which malware detected actions are triggered within the assessment computer. These malware detected actions may be similar to those previously described in relation to the client computer file. In addition, an entry specifying the malware scan fail of that computer file may also be added to the database. At step 62 an access denied result is returned to the client computer.

[0028] If the test at step 50 did not indicate that a computer file was being returned for scan, then step 64 serves to compare the file details being passed to the server with the database entries of computer files that have already been scanned. The computer file may be uniquely identified by its filename and its checksum. If at step 66 a match within the database is detected, then step 68 determines whether or not this entry indicates that access should be allowed to that computer file. If access is not to be allowed, then processing proceeds to step 60. If access is allowed, then processing proceeds to step 70 at which a pass result is returned to the client computer.

[0029] If the test at step 66 indicated that a match was not found within the database, then step 72 serves to return a remote scan required result to the client computer in order to trigger the client computer to return a copy of the computer file to the assessment computer for scanning at step 52.

[0030] Figure 4 illustrates a process that may run on the assessment computer as a background task. At step 74 the assessment computer waits to receive a lock down trigger message from the anti-virus provider. If such a message is received, then step 76 serves to activate a lock down mode in the assessment computer. The lock down mode can switch on a user predetermined set of measures intended to provide a higher degree of security, normally at the cost of at least some

Trend Micro, Inc.
EX1007-708

functionality. As an example, access to e-mail attachments or VBS files as a class may be denied. These are known to be particular vulnerabilities.

[0031] At step 78 the assessment computer waits for a user command to cancel the lock down mode. When such a command is received, then step 80 serves to cancel the lock down mode.

[0032] Figure 5 schematically illustrates the data that may be stored for a particular computer file within the database of previously conducted scan results held by the assessment computer 24. For an individual computer file, its filename, the originating computer for which the scan of that file was first conducted, the path to the file, the checksum value for the file, an allowed access flag and a persistence flag are all stored. The allowed access flag may be used to indicate whether or not that file passed its scan result. The allowed access flag may also be used as a powerful tool for switching off or on access to individual files or classes of files by an administrator. The persistence flag controls how the entry is flushed from the database on a regular interval, such as when new virus definition data is received. It will be appreciated that other fields could be added to the database relating to the particular file as required.

[0033] Figure 6 schematically illustrates a general purpose computer 200 of the type that may be used to implement the above techniques. The general purpose computer 200 includes a central processing unit 202, a random access memory 204, a read only memory 206, a hard disk drive 208, a display driver 210 and display 212, a user input/output circuit 214 and keyboard 216 and mouse 218 and a network interface unit 220 all connected via a common bus 222. In operation the central processing unit 202 executes program instructions stored within the random access memory 204, the read only memory 206 or the hard disK drive 208. The working memory is provided by the random access memory 204. The program instructions could take a variety of forms depending on the precise nature of the computer 200 and the programming language being used. The results of the processing are displayed to a user upon the display 212 driven by the display driver 210. User inputs for controlling the general purpose computer 200 are received from the keyboard 216 and the mouse 218 via the user input/output circuit 214. Communication with other computers, such as exchanging e-mails, downloading files or providing internet or other network access, is achieved via the network interface unit 220.

[0034] It will be appreciated that the general purpose computer 200 operating under control of a suitable computer program may perform the above described techniques and provide apparatus for performing the various tasks described. The general purpose computer 200 also executes the method described previously. The computer program product could take the form of a recordable medium bearing the computer program, such as a floppy disk, a compact disk or other recordable medium. Alternatively, the computer program could be dynamically downloaded via the network interface unit 220.

[0035] It will be appreciated that the general purpose computer 200 is only one example of the type of computer architecture that may be employed to carry out the above described techniques. Alternative architectures are envisaged and are capable of use with the above described techniques.

Claims

1. A computer program product for controlling a computer to detect malware, said computer program product comprising:

detecting logic operable to detect a file access request to a computer file by a requesting computer;
file access clearance request generating logic operable to generate a file access clearance request including data identifying said computer file;
file access clearance request transmitting logic operable to transmit said file access clearance request from said requesting computer to an assessment computer responsible for assessment of whether said computer file contains malware;
file access clearance response receiving logic operable to receive at said requesting computer a file access clearance response from said assessment computer; and
file access permitting logic operable if said file access clearance response indicates said computer file does not contain malware, to permit said file access request by said requesting computer.

2. A computer program product as claimed in claim 1, wherein said data identifying said computer file includes a checksum value calculated from said computer file.

3. A computer program product as claimed in claims 1 and 2, wherein said data identifying said computer file includes one or more of a filename of said computer file, data identifying said requesting computer and a storage location of said computer file.

4. A computer program product as claimed in claims 1, 2 and 3, wherein if said file access clearance response indicates a scan of said computer file is required by said assessment computer, then computer file transmitting logic is operable to transmit said computer file from said requesting computer to said assessment computer.

5. A computer program product as claimed in any

Trend Micro, Inc.
EX1007-709

one of the preceding claims, wherein if said file access clearance response indicates access to said computer file is denied, then triggering a denied access response in said requesting computer.

6. A computer program product as claimed in any one of the preceding claims, wherein a plurality of requesting computers share access to an assessment computer for determining whether file access requests by those requesting computers should be denied.

7. A computer program product for controlling a computer to detect malware, said computer program product comprising:

file access request receiving logic operable to receive at an assessment computer a file access clearance request from a requesting computer, said file access clearance request including data identifying a computer file to be accessed by said requesting computer;
file access clearance response generating logic operable in dependence upon said data identifying said computer file to determine if said computer file has previously been assessed as not containing malware and to generate a file access clearance response; and
file access clearance response transmitting logic operable to transmit said file access clearance response to said requesting computer.

8. A computer program product as claimed in claim 7, wherein said data identifying said computer file includes a checksum value calculated from said computer file.

9. A computer program product as claimed in any one of claims 7 and 8, wherein said data identifying said computer file includes one or more of a filename of said computer file, data identifying said requesting computer and a storage location of said computer file.

10. A computer program product as claimed in any one of claim 7, 8 and 9, wherein if said file access clearance response indicates a scan of said computer file is required by said assessment computer, then computer file receiving logic is operable to receive at said assessment computer said computer file from said requesting computer and performing a malware scan of said computer file.

11. A computer program product as claimed in any one of claims 7 to 10, wherein if said file access clearance response indicates access to said computer file is denied, then triggering a denied access response in said assessment computer.

12. A computer program product as claimed in any one of claims 7 to 11, wherein said assessment computer stores a database of computer files previously assessed as to whether they contain malware.

13. A computer program product as claimed in claim 12, wherein said database includes for each computer file fields specifying one or more of a filename of said computer file, data identifying said requesting computer and a storage location of said computer file, a checksum value calculated from said computer file, an access flag indicating whether access to said computer file is denied and a persistence flag indicating whether entries relating to said computer file should be purged from said database during purge operations.

14. A computer program product as claimed in any one of claims 7 to 13, wherein said assessment computer is operable in at least a higher level security mode and a lower level security mode, said assessment computer serving to deny access to greater range of computer files when operating in said higher level security mode compared with said lower level security mode.

15. A computer program product as claimed in claim 14, wherein said assessment computer is triggered to change from said lower level security mode to said higher level security mode by a lock down trigger message received at said assessment computer from a remote computer.

16. A computer program product as claimed in any one of claims 7 to 15, wherein a plurality of requesting computers share access to an assessment computer for determining whether file access requests by those requesting computers should be denied.

17. A computer program product for controlling a computer to detect malware, said computer program product comprising:

file access request detecting logic operable to detect a file access request to a computer file by a requesting computer;
file access clearance request generating logic operable to generate a file access clearance request including data identifying said computer file;
file access clearance request transmitting logic operable to transmit said file access clearance request from said requesting computer to an assessment computer responsible for assessment of whether said computer file contains malware;
file access clearance request receiving logic

Trend Micro, Inc.
EX1007-710

operable to receive at said assessment computer said file access clearance request from a requesting computer;
file access clearance response generating logic operable in dependence upon said data identifying said computer file to determine if said computer file has previously been assessed as not containing malware and to generate a file access clearance response;
file access clearance response transmitting logic operable to transmit said file access clearance response to said requesting computer;
file access clearance response receiving logic operable to receive at said requesting computer said file access clearance response from said assessment computer; and
file access permitting logic operable if said file access clearance response indicates said computer file does not contain malware to permit said file access request by said requesting computer.

18. A computer program product as claimed in claim 17, wherein said data identifying said computer file includes a checksum value calculated from said computer file.

19. A computer program product as claimed in claims 17 and 18, wherein said data identifying said computer file includes one or more of a filename of said computer file, data identifying said requesting computer and a storage location of said computer file.

20. A computer program product as claimed in any one of claims 17 to 19, wherein if said file access clearance response indicates a scan of said computer file is required by said assessment computer, then computer file transmitting logic is operable to transmit said computer file from said requesting computer to said assessment computer, receiving at said assessment computer said computer file from said requesting computer and performing a malware scan of said computer file.

21. A computer program product as claimed in any one of claims 17 to 20, wherein if said file access clearance response indicates access to said computer file is denied, then triggering a denied access response in said assessment computer.

22. A computer program product as claimed in any one of claims 17 to 21, wherein if said file access clearance response indicates access to said computer file is denied, then triggering a denied access response in said requesting computer.

23. A computer program product as claimed in any one of claims 17 to 22, wherein said assessment computer stores a database of computer files previously assessed as to whether they contain malware.

24. A computer program product as claimed in claim 23, wherein said database includes for each computer file fields specifying one or more of a filename of said computer file, data identifying said requesting computer and a storage location of said computer file, a checksum value calculated from said computer file, an access flag indicating whether access to said computer file is denied and a persistence flag indicating whether entries relating to said computer file should be purged from said database during purge operations.

25. A computer program product as claimed in any one of claims 17 to 25, wherein said assessment computer is operable in at least a higher level security mode and a lower level security mode, said assessment computer serving to deny access to greater range of computer files when operating in said higher level security mode compared with said lower level security mode.

26. A computer program product as claimed in claim 25, wherein said assessment computer is triggered to change from said lower level security mode to said higher level security mode by a lock down trigger message received at said assessment computer from a remote computer.

28. A computer program product as claimed in any one of claims 17 to 27, wherein a plurality of requesting computers share access to an assessment computer for determining whether file access requests by those requesting computers should be denied.

29. A method of detecting malware, said method comprising the steps of:

detecting a file access request to a computer file by a requesting computer;
generating a file access clearance request including data identifying said computer file;
transmitting said file access clearance request from said requesting computer to an assessment computer responsible for assessment of whether said computer file contains malware;
receiving at said requesting computer a file access clearance response from said assessment computer; and
if said file access clearance response indicates said computer file does not contain malware, then permitting said file access request by said requesting computer.

Trend Micro, Inc.
EX1007-711

30. A method as claimed in claim 29, wherein said data identifying said computer file includes a checksum value calculated from said computer file.

31. A method as claimed in claims 29 and 30, wherein said data identifying said computer file includes one or more of a filename of said computer file, data identifying said requesting computer and a storage location of said computer file.

32. A method as claimed in any one of claims 29 to 31, wherein if said file access clearance response indicates a scan of said computer file is required by said assessment computer, then transmitting said computer file from said requesting computer to said assessment computer.

33. A method as claimed in any one of claims 29 to 31, wherein if said file access clearance response indicates access to said computer file is denied, then triggering a denied access response in said requesting computer.

34. A method as claimed in any one of claims 29 to 33, wherein a plurality of requesting computers share access to an assessment computer for determining whether file access requests by those requesting computers should be denied.

35. A method of detecting malware, said method comprising the steps of:

receiving at an assessment computer a file access clearance request from a requesting computer, said file access clearance request including data identifying a computer file to be accessed by said requesting computer;
in dependence upon said data identifying said computer file determining if said computer file has previously been assessed as not containing malware and generating a file access clearance response; and
transmitting said file access clearance response to said requesting computer.

36. A method as claimed in claim 35, wherein said data identifying said computer file includes a checksum value calculated from said computer file.

37. A method as claimed in claims 35 and 36, wherein said data identifying said computer file includes one or more of a filename of said computer file, data identifying said requesting computer and a storage location of said computer file.

38. A method as claimed in any one of claims 35 to 37, wherein if said file access clearance response indicates a scan of said computer file is required by said assessment computer, then receiving at said assessment computer said computer file from said requesting computer and performing a malware scan of said computer file.

39. A method as claimed in any one of claims 35 to 38, wherein if said file access clearance response indicates access to said computer file is denied, then triggering a denied access response in said assessment computer.

40. A method as claimed in any one of claims 35 to 39, wherein said assessment computer stores a database of computer files previously assessed as to whether they contain malware.

41. A method as claimed in claim 40, wherein said database includes for each computer file fields specifying one or more of a filename of said computer file, data identifying said requesting computer and a storage location of said computer file, a checksum value calculated from said computer file, an access flag indicating whether access to said computer file is denied and a persistence flag indicating whether entries relating to said computer file should be purged from said database during purge operations.

42. A method as claimed in any one of claims 35 to 41, wherein said assessment computer is operable in at least a higher level security mode and a lower level security mode, said assessment computer serving to deny access to greater range of computer files when operating in said higher level security mode compared with said lower level security mode.

43. A method as claimed in any one of claims 35 to 42, wherein said assessment computer is triggered to change from said lower level security mode to said higher level security mode by a lock down trigger message received at said assessment computer from a remote computer.

44. A method as claimed in any one of claims 35 to 43, wherein a plurality of requesting computers share access to an assessment computer for determining whether file access requests by those requesting computers should be denied.

45. A method of detecting malware, said method comprising the steps of:

detecting a file access request to a computer file by a requesting computer;
generating a file access clearance request including data identifying said computer file;
transmitting said file access clearance request

Trend Micro, Inc.
EX1007-712

from said requesting computer to an assessment computer responsible for assessment of whether said computer file contains malware;
receiving at said assessment computer said file access clearance request from a requesting computer;
in dependence upon said data identifying said computer file determining if said computer file has previously been assessed as not containing malware and generating a file access clearance response;
transmitting said file access clearance response to said requesting computer;
receiving at said requesting computer said file access clearance response from said assessment computer; and
if said file access clearance response indicates said computer file does not contain malware, then permitting said file access request by said requesting computer.

46. A method as claimed in claim 45, wherein said data identifying said computer file includes a checksum value calculated from said computer file.

47. A method as claimed in claims 45 and 46, wherein said data identifying said computer file includes one or more of a filename of said computer file, data identifying said requesting computer and a storage location of said computer file.

48. A method as claimed in any one of claims 45 to 47, wherein if said file access clearance response indicates a scan of said computer file is required by said assessment computer, then transmitting said computer file from said requesting computer to said assessment computer, receiving at said assessment computer said computer file from said requesting computer and performing a malware scan of said computer file.

49. A method as claimed in any one of claims 45 to 48, wherein if said file access clearance response indicates access to said computer file is denied, then triggering a denied access response in said assessment computer.

50. A method as claimed in any one of claims 45 to 49, wherein if said file access clearance response indicates access to said computer file is denied, then triggering a denied access response in said requesting computer.

51. A method as claimed in any one of claims 45 to 50, wherein said assessment computer stores a database of computer files previously assessed as to whether they contain malware.

52. A method as claimed in claim 51, wherein said database includes for each computer file fields specifying one or more of a filename of said computer file, data identifying said requesting computer and a storage location of said computer file, a checksum value calculated from said computer file, an access flag indicating whether access to said computer file is denied and a persistence flag indicating whether entries relating to said computer file should be purged from said database during purge operations.

53. A method as claimed in any one of claims 45 to 52, wherein said assessment computer is operable in at least a higher level security mode and a lower level security mode, said assessment computer serving to deny access to greater range of computer files when operating in said higher level security mode compared with said lower level security mode.

54. A method as claimed in claim 53, wherein said assessment computer is triggered to change from said lower level security mode to said higher level security mode by a lock down trigger message received at said assessment computer from a remote computer.

55. A method as claimed in any one of claims 45 to 54, wherein a plurality of requesting computers share access to an assessment computer for determining whether file access requests by those requesting computers should be denied.

56. Apparatus for detecting malware, said apparatus comprising:

a detector operable to detect a file access request to a computer file by a requesting computer;
a file access clearance request generator operable to generate a file access clearance request including data identifying said computer file;
a file access clearance request transmitter operable to transmit said file access clearance request from said requesting computer to an assessment computer responsible for assessment of whether said computer file contains malware;
a file access clearance response receiver operable to receive at said requesting computer a file access clearance response from said assessment computer; and
a file access permission unit operable if said file access clearance response indicates said computer file does not contain malware, to permit said file access request by said requesting

Trend Micro, Inc.
EX1007-713

computer.

57. Apparatus as claimed in claim 56, wherein said data identifying said computer file includes a checksum value calculated from said computer file.

58. Apparatus as claimed in claims 56 and 57, wherein said data identifying said computer file includes one or more of a filename of said computer file, data identifying said requesting computer and a storage location of said computer file.

59. Apparatus as claimed in any one of claims 56 to 58, wherein if said file access clearance response indicates a scan of said computer file is required by said assessment computer, then a computer file transmitter is operable to transmit said computer file from said requesting computer to said assessment computer.

60. Apparatus as claimed in any one of claims 56 to 59, wherein if said file access clearance response indicates access to said computer file is denied, then triggering a denied access response in said requesting computer.

61. Apparatus as claimed in any one of claims 56 to 60, wherein a plurality of requesting computers share access to an assessment computer for determining whether file access requests by those requesting computers should be denied.

62. Apparatus for controlling a computer to detect malware, said apparatus comprising:

a file access request receiver operable to receive at an assessment computer a file access clearance request from a requesting computer, said file access clearance request including data identifying a computer file to be accessed by said requesting computer;
a file access clearance response generator operable in dependence upon said data identifying said computer file to determine if said computer file has previously been assessed as not containing malware and to generate a file access clearance response; and
a file access clearance response transmitter operable to transmit said file access clearance response to said requesting computer.

63. Apparatus as claimed in claim 62, wherein said data identifying said computer file includes a checksum value calculated from said computer file.

64. Apparatus as claimed in claims 62 and 63, wherein said data identifying said computer file includes one or more of a filename of said computer file, data identifying said requesting computer and a storage location of said computer file.

65. Apparatus as claimed in any one of claims 62 to 64, wherein if said file access clearance response indicates a scan of said computer file is required by said assessment computer, then a computer file receiver is operable to receive at said assessment computer said computer file from said requesting computer and performing a malware scan of said computer file.

66. Apparatus as claimed in any one of claims 62 to 65, wherein if said file access clearance response indicates access to said computer file is denied, then triggering a denied access response in said assessment computer.

67. Apparatus as claimed in any one of claims 62 to 66, wherein said assessment computer stores a database of computer files previously assessed as to whether they contain malware.

68. Apparatus as claimed in claim 67, wherein said database includes for each computer file fields specifying one or more of a filename of said computer file, data identifying said requesting computer and a storage location of said computer file, a checksum value calculated from said computer file, an access flag indicating whether access to said computer file is denied and a persistence flag indicating whether entries relating to said computer file should be purged from said database during purge operations.

69. Apparatus as claimed in any one of claims 62 to 68, wherein said assessment computer is operable in at least a higher level security mode and a lower level security mode, said assessment computer serving to deny access to greater range of computer files when operating in said higher level security mode compared with said lower level security mode.

70. Apparatus as claimed in claim 69, wherein said assessment computer is triggered to change from said lower level security mode to said higher level security mode by a lock down trigger message received at said assessment computer from a remote computer.

71. Apparatus as claimed in any one of claims 62 to 70, wherein a plurality of requesting computers share access to an assessment computer for determining whether file access requests by those requesting computers should be denied.

72. Apparatus for controlling a computer to detect

Trend Micro, Inc.
EX1007-714

malware, said apparatus comprising:

a file access request detector operable to detect a file access request to a computer file by a requesting computer;
a file access clearance request generator operable to generate a file access clearance request including data identifying said computer file;
a file access clearance request transmitter operable to transmit said file access clearance request from said requesting computer to an assessment computer responsible for assessment of whether said computer file contains malware;
a file access clearance request receiver operable to receive at said assessment computer said file access clearance request from a requesting computer;
a file access clearance response generator operable in dependence upon said data identifying said computer file to determine if said computer file has previously been assessed as not containing malware and to generate a file access clearance response;
a file access clearance response transmitter operable to transmit said file access clearance response to said requesting computer;
a file access clearance response receiver operable to receive at said requesting computer said file access clearance response from said assessment computer; and
a file access permission unit operable if said file access clearance response indicates said computer file does not contain malware to permit said file access request by said requesting computer.

73. Apparatus as claimed in claim 72, wherein said data identifying said computer file includes a checksum value calculated from said computer file.

74. Apparatus as claimed in claims 72 and 73, wherein said data identifying said computer file includes one or more of a filename of said computer file, data identifying said requesting computer and a storage location of said computer file.

75. Apparatus as claimed in any one of claims 72 to 74, wherein if said file access clearance response indicates a scan of said computer file is required by said assessment computer, then a computer file transmitter is operable to transmit said computer file from said requesting computer to said assessment computer, receiving at said assessment computer said computer file from said requesting computer and performing a malware scan of said computer file.

76. Apparatus as claimed in any one of claims 72 to 75, wherein if said file access clearance response indicates access to said computer file is denied, then triggering a denied access response in said assessment computer.

77. Apparatus as claimed in any one of claims 72 to 76, wherein if said file access clearance response indicates access to said computer file is denied, then triggering a denied access response in said requesting computer.

78. Apparatus as claimed in any one of claims 72 to 77, wherein said assessment computer stores a database of computer files previously assessed as to whether they contain malware.

79. Apparatus as claimed in claim 78, wherein said database includes for each computer file fields specifying one or more of a filename of said computer file, data identifying said requesting computer and a storage location of said computer file, a checksum value calculated from said computer file, an access flag indicating whether access to said computer file is denied and a persistence flag indicating whether entries relating to said computer file should be purged from said database during purge operations.

80. Apparatus as claimed in any one of claims 72 to 79, wherein said assessment computer is operable in at least a higher level security mode and a lower level security mode, said assessment computer serving to deny access to greater range of computer files when operating in said higher level security mode compared with said lower level security mode.

81. Apparatus as claimed in claim 80, wherein said assessment computer is triggered to change from said lower level security mode to said higher level security mode by a lock down trigger message received at said assessment computer from a remote computer.

82. Apparatus as claimed in any one of claims 72 to 81, wherein a plurality of requesting computers share access to an assessment computer for determining whether file access requests by those requesting computers should be denied

Trend Micro, Inc.
EX1007-715

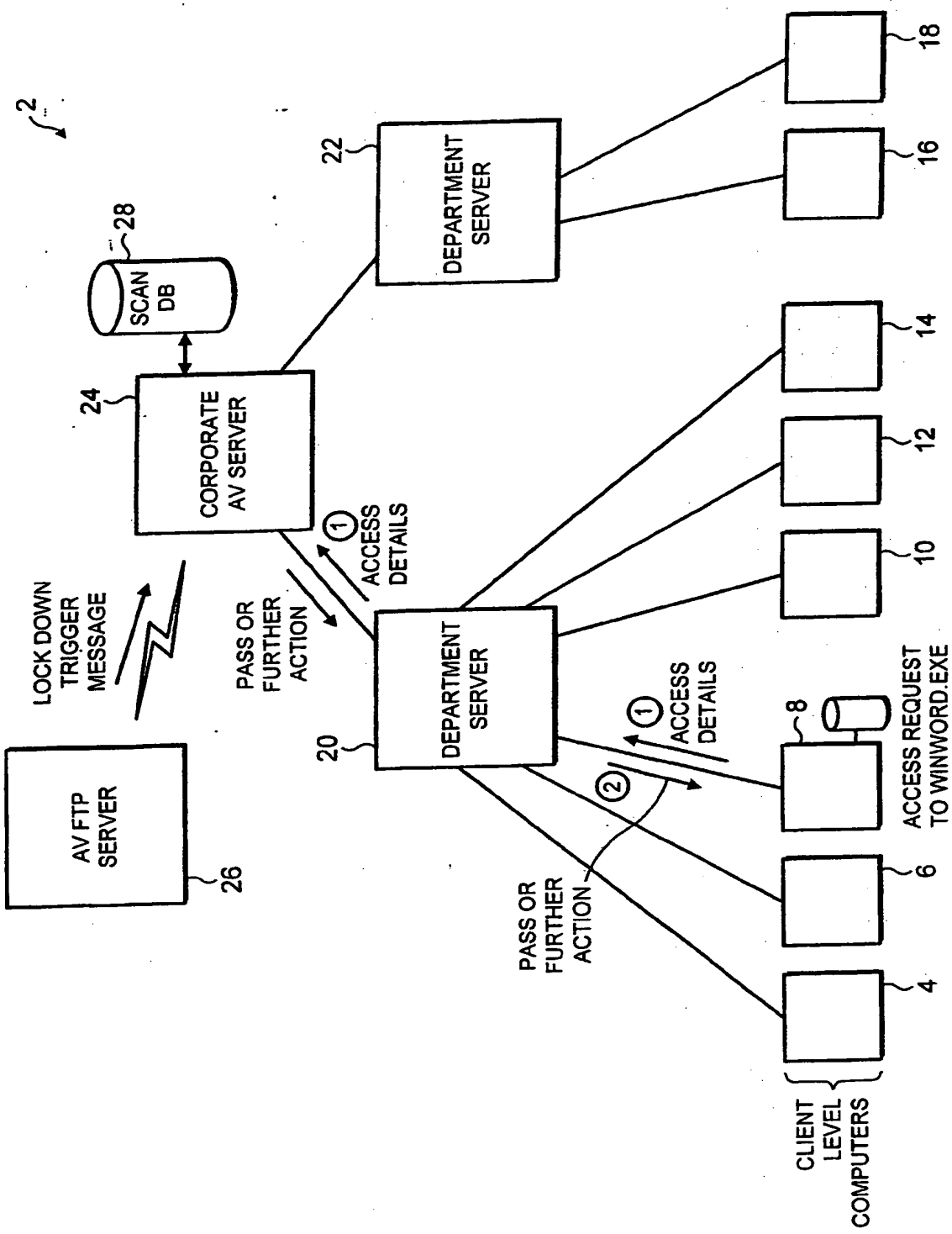


FIG. 1

Trend Micro, Inc.
EX1007-716

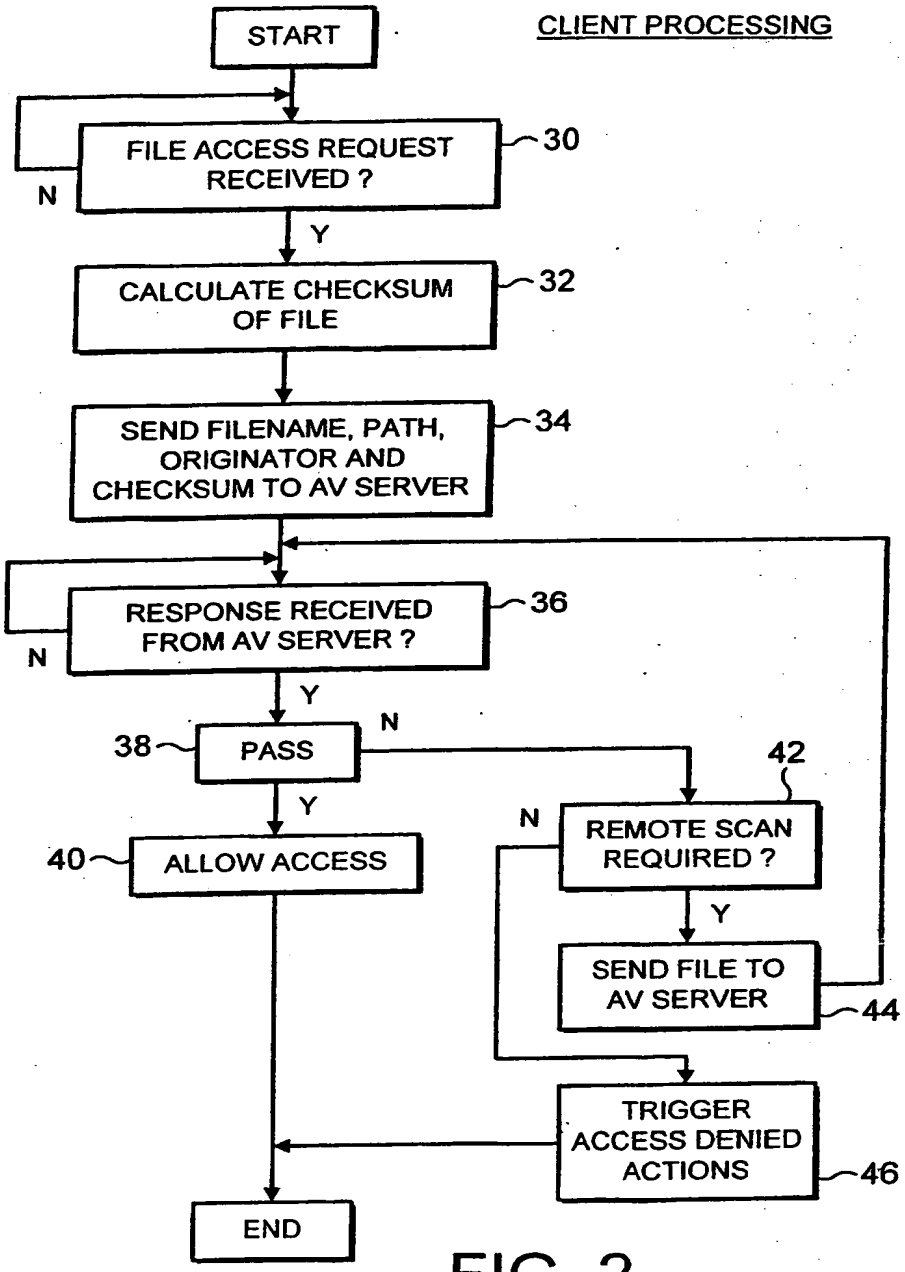


FIG. 2

Trend Micro, Inc.
EX1007-717

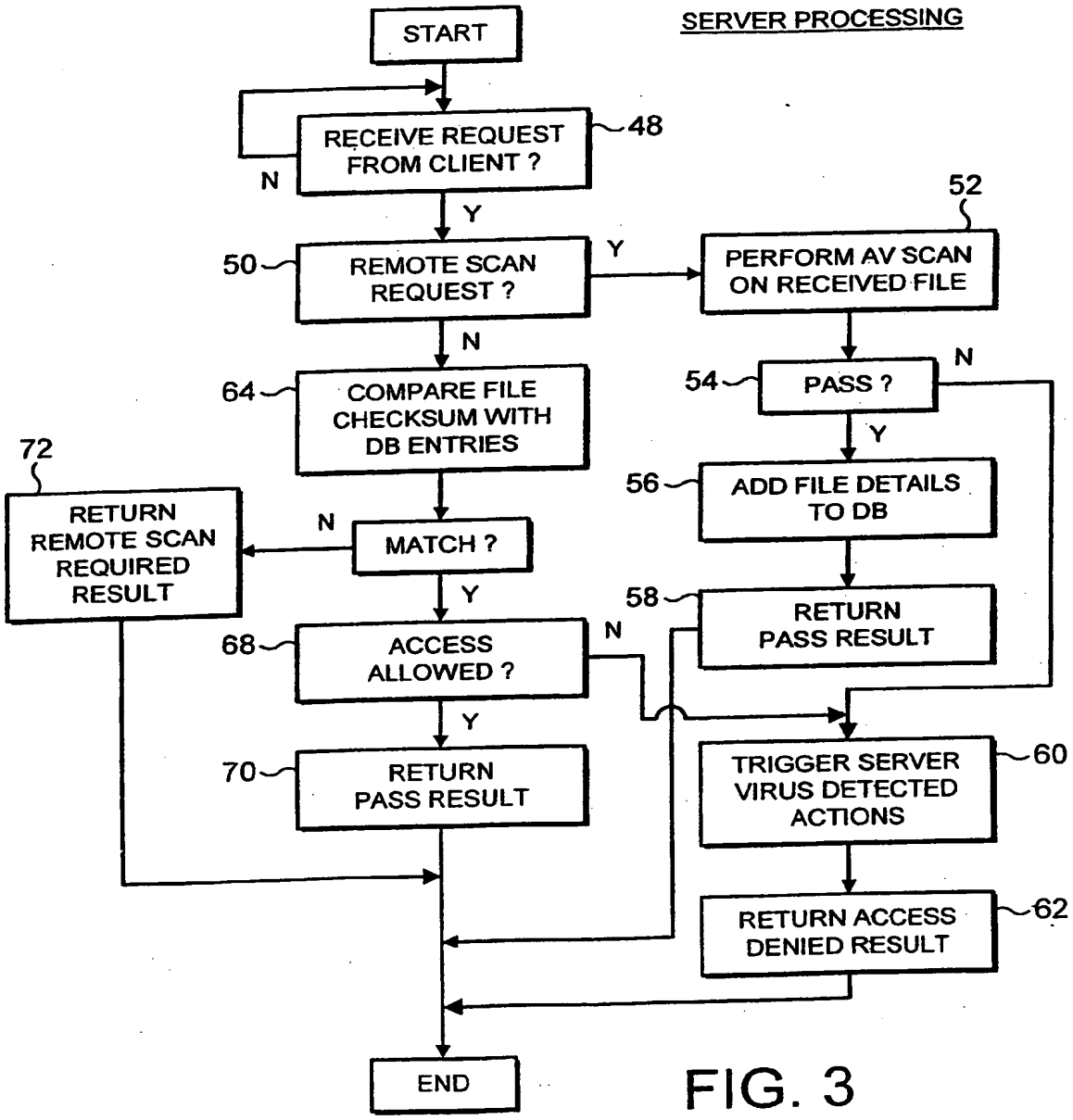


FIG. 3

Trend Micro, Inc.
EX1007-718

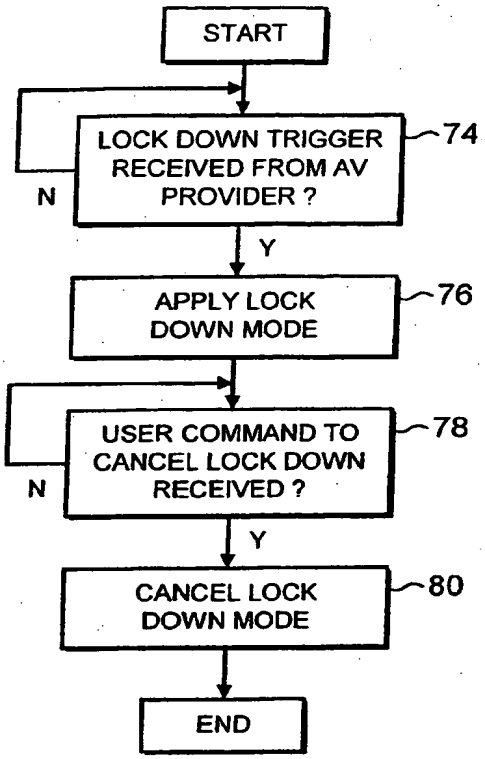


FIG. 4

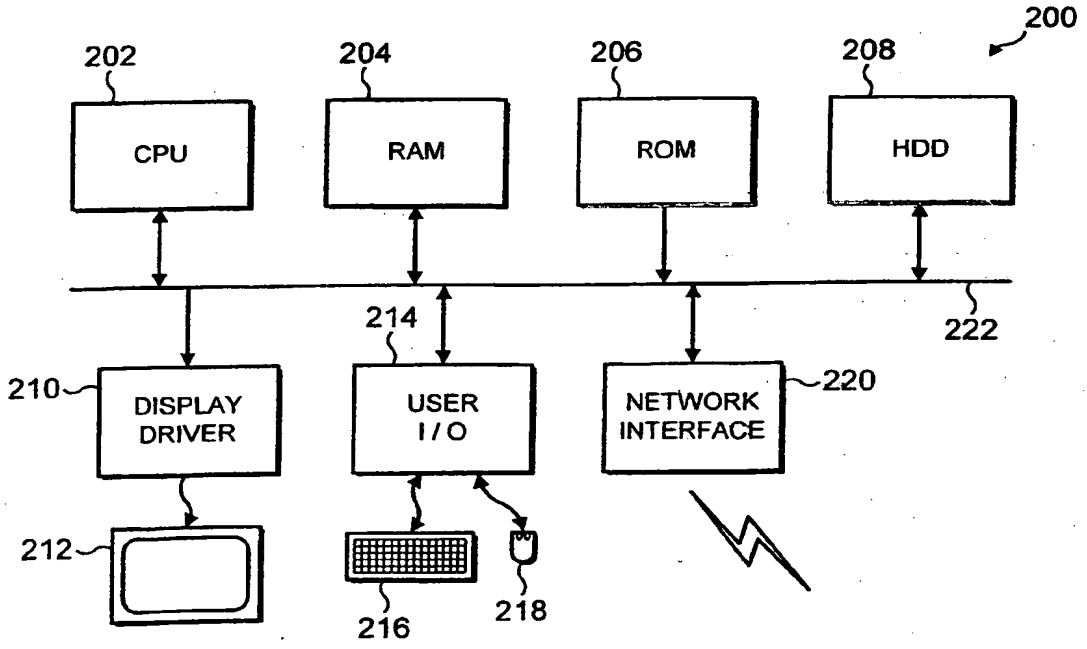


FIG. 6

Trend Micro, Inc.
EX1007-719

FILENAME	ORIGINATOR	PATH	CHECKSUM	ACCESS ALLOWED	PERSISTENT
•	•	•	•	•	•
WINWORD.EXE	CLIENT N	C:\PROGRAM FILES\ MICROSOFT OFFICE\ OFFICE	[12345678]	Y	Y
•	•	•	•	•	•

FIG. 5

Trend Micro, Inc.
EX1007-720

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☑ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 549 012 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
29.06.2005 Bulletin 2005/26

(51) Int Cl.[7]: **H04L 29/06**, G06F 1/00

(21) Application number: **03447310.8**

(22) Date of filing: **24.12.2003**

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LI LU MC NL PT RO SE SI SK TR
Designated Extension States:
AL LT LV MK

(71) Applicant: **DataCenterTechnologies N.V.**
9080 Lochristi (BE)

(72) Inventor: **De Spiegeleer, Kristof**
8301 Knokke-Heist (BE)

(74) Representative: **Bird, William Edward et al**
Bird Goen & Co.,
Klein Dalenstraat 42A
3020 Winksele (BE)

(54) Method and system for identifying the content of files in a network

(57) A method and system for performing securing and controlling of a network using content identification of files in a network having a central infrastructure and local computing devices is presented. The method comprises calculating a hash value of a new file created or received on a local computing device, transmitting the hash value to the central infrastructure, comparing the hash value with a previously determined hash value stored in a database on the central infrastructure to determine whether the file is new to the network and if the file is new to the network, checking the file content with a content identifying engine, installed and updated on the central infrastructure. Content attributes are determined for the files which allow to perform appropriate actions on the local computing devices according to policy rules.

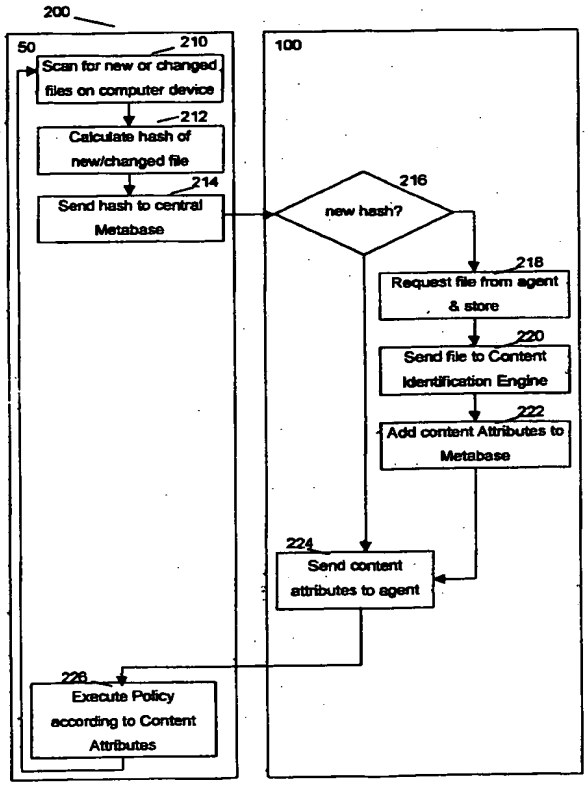


Fig. 3

EP 1 549 012 A1

BEST AVAILABLE COPY

Trend Micro, Inc.
EX1007-722

Description

Technical field of the invention

[0001] The invention relates to a method and system to control the content of computer files, e.g. containing text or graphical data. More specifically, a method and system is described for checking and managing the security status and the content of computer files on a local computing device in a network environment.

Background of the invention

[0002] In today's world, computers are widely spread. Very often, especially in business environment, they are interconnected in small or larger networks. As software and data often are an important part of the investment goods of both private persons and firms, it is important to protect single computing devices and complete networks and their workstations against attacks from viruses, trojan horses, worms and malicious software. Another problem is related to the amount of files containing undesirable content such as explicit adult content. These files are often received on local computing devices uninvited and unwanted.

[0003] To solve the security problems associated with viruses, virus protection systems, also called virus checkers, have been developed. Some examples of conventional virus checkers are Norton AntiVirus, McAfee VirusScan, PC-cillin, Kaspersky Anti-Virus. Most of these conventional virus protection software packages can be configured so that they are continuously running in the background of the computing device and providing continuous protection. These virus protection systems compare codes of new or amended software with fingerprints (e.g. parts of code introduced in files by the viruses) of well known viruses. Other virus protection systems compare codes of all data available on the computing device. This leads to the use of a significant amount of central processing unit (CPU) time, which limits the capacity of the computing device for performing other tasks. Furthermore, the working principle of these virus checkers makes these software packages work rather reactive than proactive, as the fingerprint of the virus needs to be known in order for a virus scanning program to recognise it. This implies that the database of fingerprints needs to be updated very regularly in order to be secured against relatively new viruses. Consequently, the secure state of the computer is not only depending on external factors like the accurateness with which fingerprints of new viruses are made available by the suppliers of virus protection software packages, but also on the sense of duty of the user regarding performing updates regularly. If updates are provided centrally from a server automatically, then network capacity is reduced as these virus updates must be sent to each workstation.

[0004] In a network environment, the problem of updating such a database of fingerprints becomes significantly more important, as it implies that the responsibility is put to all users, who all have to update their virus checker database. Alternatively, the virus scanning could be performed by a central server, thus limiting the updating for new fingerprints to the central server. Nevertheless this implies that a large amount of data needs to be transferred over the network on a regular basis thereby utilising large amounts of expensive network bandwidth and possibly (depending on the number of clients for the server) overloading the network or server capacity for other activities.

[0005] In order to limit the amount of CPU time used, additional techniques have been developed to speed up the virus scanning process. These very often include hashing of the content of files. Hashing is one example of application of a "one-way-function". A one-way-function is an algorithm which when applied in one direction makes the reverse direction almost impossible to perform. A one-way-function generates a value such as a hash value by a calculation on the content of a file and can uniquely fingerprint this file if the one-way-function is complex enough to avoid duplicate values from different files. The uniqueness of a hashing function depends on the type of hashing function that is used, i.e. the size of the digest that is formed and the quality of function. Good hashing functions have the fewest collisions in a table, i.e. the chance of providing the same hash value for different files is the smallest. As mentioned, this is also determined by the size of the digest, i.e. hash value, that is calculated. If e.g. a 128-bit digest is used, the number of possible different values that can be obtained is 2^{128}.

[0006] It is known to use hashing for virus checking, possibly in a network environment. Typically, a hash of an application selected to run on a local computer is calculated, a stored hash from a database on a secured computer is retrieved on the local computer, whereby the secured computer can be a secured part of the local computer or a network server, and both values are compared. If there is a match, the application is executed, if there is no match, a security action is performed. This security action comprises loading a virus scanner on the local computer. It may also comprise alerting the network administrator. Furthermore it is also known to use this for differentiating accessibility to software from different workstations and as a way of checking whether software is licensed.

[0007] It is also known to use hashing in a method of identifying rogue software on a computer system or device. The method typically is applicable in a network environment. A hash value of a software application to be executed is calculated, this hash value is transferred to a server and compared with previously stored values. One of the essential features is that the method uses a database on a server, the server being a server with a large number of clients. The database on the server thereby is built up by adding information by different cli-

Trend Micro, Inc.
EX1007-723

ents so that most software applications and their corresponding fingerprints are already stored in the database. The database is built up by checking software applications on authenticity with the owners of the application. If this is not possible, the system is also able to give a heuristic result, evaluating the occurrence of this application on local computers from other clients.

[0008] Methods for sending an electronic file by electronic mail, i.e. e-mail, including a file content and message content identifier are known. Depending on the message content identification, the message is delivered to a customer or not. The method may be used to organise e-mail delivery, but it has the disadvantage of being focussed on e-mail delivery and it does not allow to secure all files in a network.

[0009] Monitoring of electronic mail messages, to protect a computer system for protection against virus attacks and unsolicited commercial e-mail (UCE) is also known. Such a system is preferably installed on a mail server or an Internet Service Provider and checks specific parts of e-mails by calculating a digest and comparing it with stored digest values of e-mails previously received. In this way it is determined whether the e-mail has an approved digest or whether the e-mail is UCE or contains a e-mail worm. The system has the disadvantage that it is focussed on e-mail viruses and SPAM and that it does not allow to check all data files or executable files which are possibly infected, e.g. by files copied from external memory storage means like floppy disks or CD-ROMs or by e.g. Trojan horses.

[0010] Controlling the execution of software on different workstations according to certain policy rules by a network server is known, whereby an improved computer security system is obtained, by classifying software. It is suggested that this classification can be based on several forms of data one of which is e.g. the hash values of software data. This typically is performed by the calculation of hash values of a program if it is selected for loading and execution, and comparison of the hash value with a trusted value to determine the rule of execution. The classification also may be based on a hash of the content, a digital signature, the file system or network path or the URL zone.

[0011] The above mentioned methods and systems describe the use of hashing functions to check whether software applications are authentic or to regulate the execution of software applications. Nevertheless the problem of virus scanning all new files in a network using a conventional virus scanner whereby the necessity of updating the database of fingerprints of a conventional virus scanner on every local computer is limited is not discussed.

Summary of the Invention

[0012] It is an object of the present invention to provide a system and a method for identifying the content of files new on local computing devices in a network while limiting the necessity of updating the information needed by a content identifier on every local computing device.

[0013] It is an advantage of the present invention that, if the invention is used as a virus checker, the security level is further increased as the database of fingerprints of a conventional virus scanner does not have to be updated on every local computing device.

[0014] It is a specific advantage of the present invention that the content of a file new to a network is only identified once for the whole network.

[0015] It is furthermore a specific advantage of the present invention that the total processor (CPU) processing time in the network and the amount of network traffic is reduced.

[0016] The above described object and advantages are obtained with a method and system of content identification in a network according to the present invention.

[0017] The method for identifying the content of a data file in a network environment is used for a network having at least one local computing device linked to a remaining part of the network environment including a central infrastructure. The method and system comprises calculating a reference value for a new file on one of said at least one local computing devices using a one-way-function, transmitting said calculated reference value to said central infrastructure, comparing said calculated reference value with reference values previously stored within the remaining part of the network environment.

[0018] The method further comprises, after comparing, deciding that the content of the new file is already identified if a match between said calculated reference value and a previously stored reference value is found and retrieving the corresponding content attributes; or deciding that the content of the new file is not yet identified if no match between said calculated reference value and any of the previously stored reference values is found, followed by said central infrastructure identifying the content of said new file and determining content attributes corresponding with the content of the new file and storing a copy of said content attributes, after deciding, triggering an action on said local computing device in accordance with said content attributes.

[0019] In the method for identifying the content of a data file in a network environment, the reference value may be a hash value. The reference values previously stored may be stored within the central infrastructure. In the method and system for identifying the content of a data file in a network environment, identifying the content of the new file may comprise scanning the new file for viruses using an anti-virus checker means on a central infrastructure.

[0020] The method may furthermore comprise transferring the new file from the local computing device to the central infrastructure before said identifying the content of said new file is performed. Furthermore it may comprise storing a copy of the new file on the central

Trend Micro, Inc.
EX1007-724

infrastructure after transferring the new file from the local computing device to the central infrastructure. An address of where the file is stored may be stored together with the hash value, as to be able to quickly track copies of the files stored on the central infrastructure.

[0021] In the method of the present invention, triggering an action on the local computing device in accordance with said content attributes may comprise replacement of the new file on the local computing device with a copy of a previous version of said new file. Furthermore, triggering an action on the local computing device in accordance with said content attributes may also comprise replacement of the new file on the local computing device with another version of said new file restored from the remaining part of the network environment.

[0022] The method of the present invention furthermore may comprise sharing the new file on the local computing device to the central infrastructure before said identifying the content of said new file is performed and whereby said identifying the content of said new file is performed by remotely identifying the content over the network environment. The method may comprise checking the functioning of the local agent on the local computing device.

[0023] Furthermore, triggering an action on the local computing device may be performed after transmitting the content attributes corresponding to the new file to the local computing device.

[0024] In the method for identifying the content of a data file in a network environment, identifying the content of the new file may comprise one or more of the group of scanning for adult content, scanning for Self Promotional Advertising Messages or Unsolicited Commercial E-mail (UCE) and scanning for copyrighted information. The method may further relate to a method and system for providing a content firewall, whereby one local computing device is connected to the external network, which may e.g. be the internet, and the one local computing device is also connected to the network environment formed by the remaining local computing devices. The one local computing device thus links the network environment with an external network and is the only computing device that is directly connected to sources external from the network environment. The local computing device thus acts as a content firewall as to protect the network environment from attacks originating from places in the external network. The local computing device may act as a content firewall working in a promiscuous way, i.e. whereby the local computing device acts as a content firewall that sees all traffic passing by, executes the hashing and comparing functions and contacts the agents to enforce a policy.

[0025] The method may be specifically related to a method for checking the security status of a network and its components. In this embodiment, a method for determining the security status of a data file in a network environment is used in a network having at least one local computing device linked to a remaining part of the network environment including a central infrastructure. The method comprises calculating a reference value for a new file on one of the at least one local computing devices using a one-way-function, transmitting said calculated reference value to said central infrastructure, comparing said calculated reference value with reference values previously stored within the remaining part of the network environment and after comparing, deciding that the security status of the file has already been checked if a match between the calculated reference value and a previously stored reference value is found and retrieving the corresponding security status; or deciding that the security status of the new file is not yet identified if no match between said calculated reference value and any of the previously stored reference values is found, followed by said central infrastructure checking the security status of the new file and determining the security status corresponding with the new file and storing a copy of the security status, followed by

after deciding, triggering an action on said local computing device in accordance with the security status of the new file. This action may be e.g. making the file inaccessible for the user of the local computing device and for other users in the network or restoring the infected file.

[0026] The methods described above may be triggered by an action performed on the local agent. The triggering by an action performed on the local agent may be e.g. running an application or opening a file.

[0027] The invention also relates to a method for altering a system for identifying the content of a file in a network environment according to the systems described above, the network environment comprising means for calculating a one-way function, at least one local computing device linked to a remaining part of the network environment including a central infrastructure and means for identifying the content and, the method comprising altering said means for identifying the content or said means for calculating a one-way function, scanning the remaining part of the network environment for reference values calculated with a one-way function and for each of the reference values, requesting a file that corresponds with said reference value from said network environment, sending the file to means for identifying the content, identifying the content of said file and determining content attributes corresponding with the content of the file and storing a copy of said content attributes, sending the content attributes to every local computing device containing the file and after sending; triggering an action on said local computing device in accordance with said content attributes. Altering of a system for identifying the content of a file in a network environment may be triggered by the introduction of a new one-way function to calculate reference values or may be also triggered by the updating of the means for identifying the content of the files. The method may be used to scan only part of the hashing keys in the remain-

Trend Micro, Inc.
EX1007-725

ing part of the network environment, e.g. hashing keys of files of which the content is identified after a certain date, as to minimise the actions to be performed. The date of the previous content identification may be retrieved from the content attributes.

[0028] The reference values may be hashing values. The means for identifying the content may be an anti-virus checker means, a means for scanning for adult content, a means for scanning for Self Promotional Advertising Messages or a means for scanning for copyrighted information. Triggering an action on the local computing device in accordance with said content attributes may comprise replacement of the file on the local computing device with another version of the file restored from the remaining part of the network environment or may comprise replacement of a file with a copy of a previous version of the file or may comprise putting the file in quarantine or removing the file.

[0029] The invention is also related to a computer program product for executing any of the above described methods, when executed on a network. The invention furthermore relates to a system for identifying the content of a file in a network environment, said network environment comprising at least one local computing device linked to a remaining part the network environment which includes a central infrastructure and, said remaining part including a stored database, whereby the system comprises means for calculating a reference value for a new file on said local computing device using a one-way-function, means for transmitting said calculated reference value to said central infrastructure and means for comparing said calculated reference value with previously stored reference values from the database. The system furthermore comprises means for deciding whether the content of the new file is already identified based on comparison of said calculated reference value and reference values previously stored within the remaining part, means located on the central infrastructure, for identifying the content of the new file and as to assign content attributes if the new file has not been identified yet and means for storing said content attributes within the remaining part, and means for triggering an action on said local computing device in accordance with content attributes for said new file.

[0030] In the system according to the present invention, the means for identifying the content of a file may comprise an anti-virus checker means on said central infrastructure. Furthermore, means for storing a copy of the new file within the remaining part. The means for identifying the content of a file may comprise one or more of the group of means for scanning for adult content, scanning for Self Promotional Advertising Messages and scanning for copyrighted information.

[0031] The invention may also relate to a machine readable data storage device, storing the computer program product for executing any of the above described methods, when executed on a network. Furthermore, the invention may also relate to the transmission of the computer program product for executing any of the above described methods.

[0032] Although there has been constant improvement, change and evolution of methods of virus scanning and content identification of data files, the present concepts are believed to represent substantial new and novel improvements, including departures from prior practices, resulting in the provision of more efficient, stable and reliable methods of this nature.

[0033] These and other characteristics, features and advantages of the present invention will become apparent from the following detailed description, taken in conjunction with the accompanying drawings, which illustrate, by way of example, the principles of the invention. This description is given for the sake of example only, without limiting the scope of the invention. The reference figures quoted below refer to the attached drawings.

Brief description of the drawings

[0034]

Fig. 1 is a schematic representation of a computer network

Fig. 2 is a schematic representation of a central infrastructure and its basic software components

Fig. 3 is a schematic representation of a local agent-driven content identification process.

Fig. 4 is a schematic representation of a metabase-driven content identification process.

Fig. 5 is a schematic representation of a computer network to which the content firewall system and method can be applied.

[0035] In the different figures, the same reference figures refer to the same or analogous elements.

Description of illustrative embodiments

[0036] The present invention will be described with respect to particular embodiments and with reference to certain drawings but the invention is not limited thereto but only by the claims. The drawings described are only schematic and are non-limiting. In the drawings, the size of some of the elements may be exaggerated and not drawn on scale for illustrative purposes. Where the term "comprising" is used in the present description and claims, it does not exclude other elements or steps.

[0037] Furthermore, the terms first, second, third and the like in the description and in the claims, are used for distinguishing between similar elements and not necessarily for describing a sequential or chronological order. It is to be understood that the terms so used are interchangeable under appropriate circumstances and that the embodiments of the invention described herein are capable of operation in other sequences than described or illustrated herein.

[0038] In this description, the terms "file", "program",

Trend Micro, Inc.
EX1007-726

"computer file", "computer program", "data file" and "data" are used interchangeably, and any one use may imply the other terms, according to the context used. The terms "hash" and "hashing" will be used as examples of the application of one-way-functions but the present invention is not limited to a particular form of one-way-function.

[0039] The term "computing device" should be interpreted widely to include any device capable of carrying out computations and/or executing algorithms. A computing device may be any of a laptop, workstation, personal computer, PDA, smart phone, router, network printer or any other device which has a processor and can be connected to a network such as e.g. faxing devices or copiers or any dedicated electronic device such as a so-called "hardware firewall" or a modem.

[0040] The method and system to secure and control a network by identifying the content of each new file in the network can be used on any type of network. This may be a private network which may be a virtual private network, a local area network (LAN) or a wide area network (WAN). This may also be within a part of a public wide area network such as the internet. If a part of a public wide area network is used, this may be performed by remotely providing the method and system for identifying the content of each file by a service provider using an ASP or XSP business model, wherein the central infrastructure is provided to a paying client operating a local computing device. An exemplary network 10 is shown in figure 1, showing several local computing devices 50a, 50b, ... , 50i and a central infrastructure 100, also called a server. The number of local computing devices 50 connected to the network 10 is not limiting for the method of securing and controlling a network 10 according to the current invention. In business environment this number of local computing devices 50 typically ranges from a few to a few thousands. The method and system for identifying the content of each new file present in the network 10 may be used with many different operating systems such as Microsoft DOS, Apple Macintosh OS, OS/2, Unix, DataCenter-Technologies' Operating Systems, ...

[0041] In order to provide a quick method of securing and determining content identification of files, the method and system according to the present invention will determine hash values of new files present on the local computing devices 50, compare them with previously stored hash values and file information on a central server and determine the content of files new to the network 10 using a content identifying engine on the central infrastructure 100. The content attributes describing the content of a new file are then send to the local computing device 50 where an appropriate action is performed. It is also possible that the content attributes are not sent to the local computing device 50 but that the appropriate action is triggered from the central infrastructure 100. New files typically are files wherein new content has been generated on a local computing device 50 or when an external file has been received. The wording "file" may refer to data as well as to software applications, also called software.

[0042] The central infrastructure 100 contains a database, also called metabase 110, which contains a record for every hash value that is calculated for a file that already exists on one of the local computing devices 50. Besides the hash value, this record also contains a number of other fields. In these fields, file source information is stored. The file source information corresponding with a specific hash value includes the file name, a list of local computing devices 50 where the files that correspond to this hash value are residing on, including the path to the file on the file system of the local computing devices 50 and the date of last modification. An example of file source information for a specific file is given in Table 1.

Table 1

Filename	Myexampleword.doc
Path	c:\data\
Assetname	PCmarketing001
ModDate	23/4/2002

[0043] In a further field, a list of content attributes that identifies the type of content that is enclosed by the file is stored. The content attributes can e.g. refer to a file that contains a virus, a file that is a copyrighted MP3 audio file, a file that is a copyrighted video file, a file that is a picture, a file that is a picture that might contain adult content, a file that is a Self Promotional Advertising Message (SPAM), a file that is a HOAX, a file containing explicit lyrics or a file containing pieces of executable code. This list is not limiting.

[0044] The central infrastructure 100 furthermore contains a content identification engine 120. This can be a software application 130 or a set of software applications 130a, 130b, 130c, 130d, ... that use the content of a file to determine which type of content the file contains. These software applications may be various :

- a virus scanner : this is a piece of software that scans the content of the presented file and compares it with a database of known fingerprints of viruses. This can be any conventional virus scanning software like e.g. Norton anti-virus by Symantec Corporation, McAfee by Network Associates Technologies Inc., PC-cillin by Trend Micro, Kapersky Anti-Virus by Kaspersky Lab, F-secure Anti-Virus by F-Secure Corporation, ...
- an adult content in pictures scanner : this is a piece of software that scans the content of the presented file for the presence of shading, colors, textures that might represent adult content. Scanning pictures for adult content is already known. Content adult can e.g. be determined by the amount of nude that is

Trend Micro, Inc.
EX1007-727

shown. Skin tones have hue-saturation values that are in a specific range. Therefore, if an image is scanned, it is possible to determine the amount of pixels having a skin tone character and to compare it with the total number of pixels. The ratio of skin tone pixels to the total number of pixels allows to determine a ratio of possible adult content in an image. Thresholds often are introduced so that images can be classified according to their possible adult content. In similar way, video images can be categorised, whereby the video is split into its different frames and wherein the images are categorised according to the above method.

- A scanner for internet content ratings : A piece of software that scans objects for adult content based on the PICS, i.e. the Platform for Internet Content Selection, label system. On voluntary basis, internet content providers can provide internet objects with a PICS rating determining the adult content in the internet object. This PICS rating is stored in the meta data of the object. This data normally is not visible to the viewer of an internet object. The rating systems is well known and an example of a scanner for internet content ratings is provided in the Netscape web browser for scanning the content of web pages.
- A scanner for scanning an object for explicit lyrics which may indicate adult content. This is known for both text files and audio files. Audio files are first transferred to text files. Subsequently, the text files are scanned and compared with databases which contain explicit lyrics.
- a SPAM-engine: A piece of software that scans the content of e-mail messages for the presence of unalleged SPAM. Algorithms to recognize SPAM area already known. These are typically based on decomposing the text in an electronic mail message, associating statistics with the text using a statistical analyzer and coupling a neural network engine to the statistical analyzer to recognize unwanted messages based on statistical indicators.

Other examples of software applications that could be used in the content identification engine 120 are e.g. engines that scan for copyrighted content and that compares the content of the file to a database of copyrighted information, etc. In some adoptions, a human operator can pursue the role of content identification engine 120, where he manually tags a file with a content identification attribute. When the content identification engine 120 is activated, it will take a file from the local agent as input and produce a set of attributes that represent the detected content.

[0045] The content Identification engine 120 will thus be constructed as a piece of software aggregating the functionality of a set of third party engines.

[0046] In a further embodiment of the invention, a system and method in accordance with the above embodiment is described whereby the record corresponding with a specific hash value stored in the metabase 110 also comprises a field wherein the location of the file on the central infrastructure 100 corresponding with the hash value is stored. In this embodiment, a copy of all different files present on the local computing devices 50 in the network 10 is stored on the central infrastructure 100. So, the central infrastructure 100 of this embodiment also comprises a large amount of storing space. This preferably is a secured part of the central infrastructure 100, not directly connected to the network 10 so that these identical copies of the files present on the local computing devices 50 can be used in case the files on the local computing devices 50 are corrupted e.g. by a virus.

[0047] The hash value of the files are calculated using a hashing function. A hashing function typically is a one way function, i.e. given the digest, it is at least computationaly prohibitive to reconstruct the original data. Different types of hashing functions could be used : MD5, SHA-1 or ripemd all available from RSA Data Security Inc., haval which is designed at the University of Wollongong, snefru which is a Xerox secure hash function, etc.. The hashing functions most often used are MD5 and SHA-1. The MD5 algorithm takes as input a message of arbitrary length and produces as output a 128-bit "fingerprint" or "message digest" of the input. It is conjectured that it is computationally infeasible to produce two messages having the same message digest, or to produce any message having a given pre-specified target message digest. The MD5 algorithm is intended for digital signature applications, where a large file must be 'compressed' in a secure manner before being encrypted with a private (secret) key under a public-key cryptosystem. The MD5 algorithm is designed to be quite fast on 32-bit machines. In addition, the MD5 algorithm does not require any large substitution tables; the algorithm can be coded quite compactly. An alternative hashing function SHA-1, i.e. Secure Hashing Algorithm - 1, is a hashing algorithm generating a 160-bit hash. Newer versions of this algorithm also provide bit lengths of 256 and 512.

[0048] In the above mentioned embodiments describing a method and system to secure and/or control the network 10, a local agent is installed on the local computing device 50. The local agent is a piece of software that is running on a local computing device 50 and that performs certain algorithms and procedures. The local agent on the local computing device 50 is triggered typically in situations where new content is being generated on local computing devices 50. In order to avoid unnecessary hash value calculations and data transfer, a policy is setup to determine which actions will trigger the local agent and which actions do not trigger it. If e.g. a text document is being created, it is not necessary to check the file every time the document is saved. The policy for such a type of documents would preferably be that the document is checked e.g. if the file is both saved

Trend Micro, Inc.
EX1007-728

and closed. Some examples of actions which could trigger the local agent and thus start the content identification process are opening or receiving e-mail messages, opening or receiving e-mail attachments, running executable files, running files with dll or piff extension ... Applying this policy thus allows to prevent from continuously checking and scanning of documents leading to a limitation of the number of unnecessary hash calculations and content identification operations and thus limiting the unnecessary use of CPU time and load on the network traffic. The method and system of content identification is not limited due to the type of application in which the file is made.

[0049] The content identification process can be either triggered by the local agent on the local computing device 50 or it can be triggered by the central infrastructure 100. The latter process typically occurs in situations wherein new algorithms or tools are being used for content identification. Such new algorithms or tools can either be optimised algorithms and tools or previously uninstalled tools. Some examples of these tools, without restricting to these functions, could be virus checking, checking whether a file is a copyrighted MP3 Audio File, checking whether a file is a copyrighted Video File, checking whether a file is a picture that might contain adult content, checking whether a file is tagged as being SPAM or HOAX, checking whether a file contains explicit lyrics or checking whether a file contains copyrighted pieces of executable code. Updating of these tools may influence the status of the files and thus may in principle influence the corresponding records in the metabase 110. Therefore, depending on the type of update of the content identification means 120, it may be interesting to update the corresponding records.

[0050] In a specific embodiment, the method relates to a virus checker for a network environment. The networks 10 on which this method can be applied are the same as those described for the previous embodiments. The local agent calculates the hash value of a new file on the local computing device 50. This new file may comprise new content generated on the local computing device 50 or an external file which is received on the local computing device 50. The hash value of the new file and the corresponding file information then is sent to a central infrastructure 100, also called server, where it is compared to previously stored hash values corresponding with files that are already present on the different local computing devices 50 of the network 10. This comparison allows to check whether the file is new or not in the entire network 10. Alternatively, the hash value may also be first compared with a local database of hash values and file information corresponding with the files present on that particular local computing device 50 and subsequently, if the file has been found not yet present on the local computing device 50, the hash value and the corresponding file information may be interchanged with the central infrastructure 100 so it can be checked whether the file is new or not in the entire network 10. Although transferring the file information and the hash value of every new file only corresponds with a very small fraction of the network traffic for a conventional central virus checker, this alternative could reduce the network traffic used for virus checking even further. If a hash value has been identified as new on the network 10, the metabase agent triggers the local agent to transfer the file corresponding with the new hash value from the local computing device 50 to the central infrastructure 100. The transferring of the file may be performed in a secured way, i.e. the file may be transferred such that it cannot be influenced by a virus present at a network connection or such that, it if contains a virus, this cannot be spread over the whole network 10. To obtain this, a known secure transmission route, a tunnel and/or known session ecryption/decryption techniques may be used. A conventional virus checker, installed and updated on the central infrastructure 100 then checks the file for viruses. This can be any conventional virus checker like e.g. Norton anti-virus by Symantec Corporation, McAfee by Network Associates Technologies Inc., PC-cillin by Trend Micro, Kapersky Anti-Virus by Kaspersky Lab, F-secure Anti-Virus by F-Secure Corporation, ...

[0051] A specific advantage of the above described embodiments in the current invention is that the virus scanning software does not need to be updated on every local agent but that this is restricted to updating of the virus scanning software of the central infrastructure 100. In this way the security level of the network 10 is increased significantly as the security does not depend on the punctuality of the different users of the network 10 to update their virus scanning software. If the scanned file has no virus it will be marked in the metabase 110 as being a virus free file. If there is a virus found in a file the file will be marked as dangerous. A query will happen to the metabase 110 to find all files over the network 10 having the same corrupted hashing key. The result is a list of files with path, and assetname where the file is located. This information can be used to do actions to eleminate the danger of found viruses on all local computing devices 50, i.e. all workstations, from the complete network 10. In this way proactive virus scanning can be performed on other local computing devices 50, based on a virus detection on a first local computing device 50. Depending the policy defined for virus checking, the virus engine will inform an agent installed on the affected system to remove the file and if possible replace with either a recovered version delivered by the virus engine located on the central infrastructure 100, or a previous version of the file which didn't have the virus yet. The latter can be done easily by searching the metabase for a previous version of that file, or it can be performed by searching an uninfected version on another local computing device 50. If an uninfected version cannot be retrieved from either another local computing device 50 or the metabase residing on 100, the virus scanner should have a feature which allows it to save a

Trend Micro, Inc.
EX1007-729

new disinfected copy of the file on the central infrastructure 100. These advantages are also present for other content identifying packages.

[0052] In an alternative embodiment, if a file having a new hash value has been identified in the network 10, instead of transferring the file to the central infrastructure, the file may be automatically shared locally and a remote checker then may transfer a file-system which allows to check the file across the network 10 using the file sharing. The content tagging still is performed by the server. In order to improve security, the accessibility to the shared file is restricted to the server. Furthermore, a java applet could be transferred to the local agent to allow checking other files.

[0053] The previous embodiments are an improvement over a central virus checker which scans local computing devices 50 through the network 10. This is only possible if the local drives, e.g. C:\, D:\, ..., are shared. Besides the dangers of sharing with respect to security, the local user also easily can change the local sharing properties thereby preventing the remote checker from checking the files. This is at least partly avoided with the current invention as changing the network 10 sharing properties does not influence the operation of calculating the hash value of new files and sending it to the central infrastructure 100.

[0054] Another advantage is that it saves CPU time on the local computing device 50 as the CPU does not have to keep doing virus checking, it only has to calculate a one way function. It also saves network time : the administrating server does not have to update the virus checkers on the local computing devices 50 with virus updates, as a single central virus checker only is used and updated.

[0055] Fig. 3 shows a method 200 of the content identification process triggered by the local agent on the local computing device 50 according to the above mentioned embodiments. The different steps that occur during the process, both on a local computing device 50 and on the central infrastructure 100 are discussed.

[0056] The content identification process is based on continuously scanning for new data or applications on the local computing device 50 by the local agent. This scanning for data and applications is limited by the policy rule for determining when the local agent should be triggered, as described above. If a "new" file has been detected the method for securing and controlling the network 10 by content identification of new files is initiated. This is step 210. Method 200 then proceeds to step 212.

[0057] In step 212, a hash value of the "new" file is calculated using a hashing function like MD5 or SHA-1. This calculation is performed by using some CPU time of the local computing device. Nevertheless, the amount of CPU time used is drastically smaller than the CPU time that would be necessary if e.g. a conventional virus checker was used to check the file on the local computing device 50. Method 200 then proceeds to step 214.

[0058] In step 214, the hash value and the file source information is transferred from the local agent to the central infrastructure 100 of the network 10. If necessary, this transfer can be a secured transfer, whereby it is avoided that a virus which is positioned on a network connection changes both the file source information or the hashing key during transport of this data. Such a secured transmission can be made over a known secure transmission route, via a tunnel, or using known session ecryption/decryption techniques.

[0059] In step 216, the hash value is compared with the data already present in the metabase 110. As in the metabase 110, the hash values and file source information of all old files - i.e. every file that has been present on the network 10 and that is not "new" as described above - present in the network 10 are stored, it is possible to check whether the file already is present in the network 10. Therefore, if the hash value has been identified as new, this implies that the file is "new" for the whole network 10. If the file is new, method 200 proceeds to step 218. If the hash value is not new, this means that somewhere on a local computing device 50 in the network 10, the file does already exist. In this case, there already exists content attributes describing the content of the file. Method 200 then proceeds to step 224.

[0060] In step 218 the metabase agent triggers the local agent to transfer the file corresponding with the new hash value from the local computing device 50 to the central infrastructure 100. The transferring of the file may be performed in a secured way, i.e. the file may be transferred such that it cannot be influenced by a virus present at a network connection or such that, it if contains a virus, this cannot be spread over the whole network 10. To obtain this a known secure transmission route, a tunnel and/or known session ecryption/decryption techniques may be used. Method 200 further proceeds to step 220.

[0061] In step 220 the file is loaded in the content identification engine 120 and the file is processed. For this processing CPU time of the central infrastructure 100 is used. The content identification engine 120 can comprise, as described above, a conventional virus checker, a means for checking picture information, a means for checking SPAM, etc. This can be a repetitive action where multiple content identification engines are called in turn. Method 200 then proceeds to step 222.

[0062] In step 222 content attributes, which identify the content of the file, are determined for the file. These content attributes are then stored in the metabase 110, thus allowing to identify the status of the file if, in future operations, the file is found 'new' on another local computing device 50. Method 200 then proceeds to step 224. Depending on the embodiment used, a following step may include the storing of the file on the central infrastructure 100 and adding the path to this file to the metabase 110. This step is not shown in figure 3.

[0063] In step 224 the content attributes are sent to

Trend Micro, Inc.
EX1007-730

the local agent. Based on this content attributes, the local agent performs an appropriate action in agreement with the policy rule set for these content attributes. This is performed in step 226. This can be e.g. deleting the file if it was infected, replacing the file with a previous version which was not infected, ... In a specific embodiment, the execution of appropriate actions based on the policy rules are triggered by the agent of the metabase 110, so that step 224 can be avoided.

[0064] The content policy is a policy that determines what should be done with a file depending on the content attributes determined by the content identification engine 120. The content policy can comprise actions such as deleting the file, deleting the file and replacing it with a previous version, copying the file onto another computing device while leaving a copy on the originating computing device, moving the file onto another computing device while deleting the original file on the originating computing device, logging the presence of the file, changing the attributes of the file like hiding it or making it read-only, making the file unreadable, making the file un-executable, etc. The content policy will be executed by the local agent, e.g. when the content attributes are received from the central infrastructure 100. The content policy for that agent will be downloaded to the local computing device 50 by the agent from a central policy infrastructure.

[0065] Fig. 4 shows a method 300 of the content identification process triggered by the content identification engine 120 according to the above mentioned embodiments . The different steps that occur during the process, both on a local computing device 50 and on the central infrastructure 100 are discussed.

[0066] This process typically is used in situations where new algorithms or tools are being used for content identification. Such new algorithms or tools can either be optimised algorithms and tools or previously uninstalled tools. As mentioned earlier this may be regulated by a policy : the triggering of the content identification process may be determined by the type of new algorithms and tools that are being used for content identification.

[0067] Method 300 is initiated by change of the content identification engine 120, e.g. by providing new algorithms or tools for the content identification engine 120.

[0068] In the first step 302 of method 300 upon triggering, the metabase 110 is scanned for hash values corresponding with hashing keys. Method 300 then proceeds to step 304.

[0069] In step 304, a file that corresponds with the hashing key is requested. This file can be either requested from the central storage on the central infrastructure 100 or it can be requested from a local computing device 50. The local computing device 50 then gives permission to the central infrastructure 100 to upload the corresponding file. The path to the file corresponding with the hash value is available from the record corresponding with each hash value. If the record stores different paths all corresponding with a copy of the corresponding file, the agent on the central infrastructure 100 retrieves one copy of the file, e.g. by scanning the paths listed in the record until a local computing device 50 has been found that is at that time connected to the network 10 and that allows uploading of the file. Method 300 then proceeds to step 306.

[0070] Once the file has been retrieved, the file is sent to the content identification engine 120. This is performed in step 306. The upgraded content identification engine 120 then scans the content of the file and produces content attributes corresponding with the file. Method 300 further proceeds to step 308.

[0071] In step 308, the content attributes are stored in the metabase 110, to allow in future security steps to immediately identify the content of the files. Method 300 further proceeds to step 310.

[0072] In step 310, the content attributes are sent to every local agent that resides on a local computing device 50 whereon the corresponding file is stored. The paths can be found in the record of the corresponding hashing key stored in the metabase 110. In this step, content attributes are send to every file for which a path is mentioned in the record of the corresponding hashing key. Method 300 proceeds to step 312.

[0073] In step 312, the local agent on the corresponding local computing devices 50 executes the policy according to the content attributes and the according to the local computing device 50.

[0074] One of the major advantages of the embodiments of the invention is that a file new to the entire network 10 only needs to be scanned once. If on another local computing device 50, an identical copy of this file is used, installed, opened or saved and closed, the file will be recognised by the central infrastructure 100 as being known to the network 10, in this way avoiding the need to re-check the content of the file. This especially is advantageous if the invention is used for networks 10 having a large number of local computing devices 50.

[0075] When operating, the method and system for identifying the content of new files optionally can comprise checking 'the heartbeat' of the local agent at regular times, i.e. it can be checked whether the local agent is still running on the local computing device 50. This can avoid that a user locally shuts down the agent, thus making the local computing device 50 vulnerable. If the local agent has been shut down, the network administrator can be warned. Furthermore a warning message could be send to the local computing device 50 thereby warning the user of the local computing device 50. The network administrator could also put the local computing device 50 in quarantine so that it can not damage other local computing devices 50 in the network 10. Furthermore, the central agent can also try to rerun the local agent.

[0076] In a similar way, the method and system for identifying the content of new files optionally can check

Trend Micro, Inc.
EX1007-731

at regular times whether the local computing device 50 is still connected to the network 10. If the local computing device 50 is not connected to the network 10 anymore, the local agent may further operate, storing hashing keys of new files in a waiting list to be checked once the network connection is restored. In the mean time, the corresponding files may be put in quarantine or depending on the type of file e.g. may be prevented from being executed.

[0077] The above described embodiments may be used as a content firewall for the different computing devices connected to the external network. For every incoming/outgoing file, incoming/outgoing message or incoming/outgoing data frame, the content firewall calculates the hash, checks whether this is new, checks whether it is tagged for specific content and enforces the policy associated with the specific content.

[0078] In a further embodiment, another configuration for using the present invention as a content firewall is described. A schematic overview of a computer network wherein this method and system may be used, is shown in Fig. 5. Only one reconfigurable firewall electronic device 50, such as a local computing device which may be in the form of a dedicated reconfigurable firewall electronic device, is directly connected to an external network 400 such as e.g. the internet, and the remaining local computing devices 410, are not directly connected to the external network 400, but grouped in a network environment and only connected to the external network 400 by their connection to the electronic reconfigurable firewall device.. The external network may be any possible network available. It is a goal of the content firewall as represented by the reconfigurable electronic firewall device 50 to protect the network environment comprising the remaining local computing devices 410 from attacks originating from places and/or devices in the external network. The reconfigurable electronic firewall device 50 50 either contains a local copy of the metabase or it can use a high speed secured network to a central infrastructure 100 which is part of the internal network. This allows for fast queries through the metabase. During operation, the reconfigurable electronic firewall device 50 functioning as a content firewall performs the following actions : the hash value of incoming files or incoming messages or incoming data frames are calculated. Subsequently, the calculated hash values are compared with the metabase, which is either stored locally or by using a high speed secured network, and it is determined whether the incoming file, incoming message or incoming data frame is new. Furthermore, it is checked whether this file, this message or this data frame is tagged for specific content. Depending on the specific content, a policy is enforced which is associated with the specific content. This policy may be to let it pass through to its final destination, to drop it, to log it, to put it in quarantine, etc. This system requires sufficient CPU power, in order not to slow down the network speed noticeably.

[0079] In the case where none of the local computing devices connected to the network is equipped with a removable device, i.e. allowing for non-scanned content to be opened or executed on that device, this is a very secure and managable setup.

[0080] In another embodiment of the invention, a similar configuration for use of the present invention as a content firewall in promiscuous mode is provided. The content firewall thereby sees all traffic passing by, executes the hashing and comparing functions and contacts the agents to enforce a policy. The advantage of this approach is that there is no single point of failure and no bottleneck anymore and furthermore that still no resources are used on the local computing devices for calculating hashes. Furthermore, no bandwidth is used for contacting the central metabase. The disadvantage is that local agents need to be installed on all computing devices of the internal network.

In accordance with the above described embodiments, the present invention includes a computer program product which provides the functionality of any of the methods according to the present invention when executed on a computing device. Further, the present invention includes a data carrier such as a CD-ROM or a diskette which stores the computer product in a machine readable form and which executes at least one of the methods of the invention when executed on a computing device. Nowadays, such software is often offered on the Internet, hence the present invention includes transmitting the printing computer product according to the present invention over a local or wide area network.

Claims

1. A method for identifying the content of a file in a network environment, said network environment comprising at least one local computing device linked to a remaining part of the network environment including a central infrastructure and, the method comprising

   - calculating a reference value for a new file on one of said at least one local computing devices using a one-way-function,
   - transmitting said calculated reference value to said central infrastructure,
   - comparing said calculated reference value with reference values previously stored within the remaining part of the network environment,
   - after comparing,

     - deciding that the content of the new file is already identified if a match between said calculated reference value and a previously stored reference value is found and retrieving the corresponding content attributes; or

Trend Micro, Inc.
EX1007-732

- deciding that the content of the new file is not yet identified if no match between said calculated reference value and any of the previously stored reference values is found, followed by said central infrastructure identifying the content of said new file and determining content attributes corresponding with the content of the new file and storing a copy of said content attributes,
- after deciding, triggering an action on said local computing device in accordance with said content attributes.

2. A method according to claim 1, wherein the reference value is a hash value.

3. A method according to claim 1 or 2, wherein reference values previously stored are stored within the central infrastructure.

4. A method according to the previous claim wherein said identifying the content of said new file comprises scanning the new file for viruses using a antivirus checker means on said central infrastructure.

5. A method according to any of the previous claims furthermore comprising transferring the new file from the local computing device to the central infrastructure before said identifying the content of said new file is performed.

6. A method according to claim 5, furthermore comprising storing a copy of the new file on the central infrastructure after said transferring the new file from the local computing device to the central infrastructure

7. A method according to any previous claim wherein said triggering an action on said local computing device in accordance with said content attributes may comprise replacement of the new file on the local computing device with a copy of a previous version of said new file.

8. A method according to any previous claim, wherein said triggering an action on said local computing device in accordance with said content attributes may comprise replacement of the new file on the local computing device with another version of said new file restored from the remaining part of the network environment.

9. A method according to claim 1, 2, 3 or 4 furthermore comprising sharing the new file on the local computing device to the central infrastructure before said identifying the content of said new file is performed and whereby said identifying the content of said new file is performed by remotely identifying the content over the network environment.

10. A method according to any of the previous claims, wherein said triggering an action on said local computing device in accordance with said content attributes is performed after transmitting the content attributes corresponding to the new file to the local computing device.

11. A method according to the previous claims wherein said identifying the content of said new file comprises one or more of the group of scanning for adult content, scanning for Self Promotional Advertising Messages and scanning for copyrighted information.

12. A method according to any of claims 1 to 3 or 5, whereby one local computing device links said network environment with an external network, said one local computing device acting as a content firewall as to protect the network environment.

13. A computer program product for executing any of the methods as claimed in claims 1 to 12 when executed on a network.

14. A system for identifying the content of a file in a network environment, said network environment comprising at least one local computing device linked to a remaining part the network environment which includes a central infrastructure and, said remaining part including a stored database, whereby the system comprises :

- means for calculating a reference value for a new file on said local computing device using a one-way-function,
- means for transmitting said calculated reference value to said central infrastructure,
- means for comparing said calculated reference value with previously stored reference values from the database,

whereby the system further comprises:

- means for deciding whether the content of the new file is already identified based on comparison of said calculated reference value and reference values previously stored within the remaining part,
- means located on the central infrastructure, for identifying the content of the new file and as to assign content attributes if the new file has not been identified yet and means for storing said content attributes within the remaining part, and
- means for triggering an action on said local

Trend Micro, Inc.
EX1007-733

computing device in accordance with content attributes for said new file.

15. A system according to claim 14 wherein said means for identifying the content of a file comprise an anti-virus checker means on said central infrastructure.

16. A system according to claim 14 or 15 furthermore comprising means for storing a copy of the new file within the remaining part.

17. A system according to any of claims 14 to 16 wherein said means for identifying the content of a file comprises one or more of the group of means for scanning for adult content, scanning for Self Promotional Advertising Messages and scanning for copyrighted information.

18. A method for altering a system for identifying the content of a file in a network environment according to claims 14 to 17, said network environment comprising means for calculating a one-way function, at least one local computing device linked to a remaining part of the network environment including a central infrastructure and means for identifying the content and, the method comprising

- altering said means for identifying the content or said means for calculating a one-way function
- scanning the remaining part of the network environment for reference values calculated with a one-way function
- for each of said reference values,
    - requesting a file that corresponds with said reference value from said network environment
    - sending the file to means for identifying the content
    - identifying the content of said file and determining content attributes corresponding with the content of the file and storing a copy of said content attributes
    - sending the content attributes to every local computing device containing the file
    - after sending; triggering an action on said local computing device in accordance with said content attributes.

19. A computer program product for executing the method as claimed in claims 18 when executed on a network.

20. A machine readable data storage device storing the computer program product of any of claims 13 or 19.

21. Transmission of the computer program product of any of claims 13 or 19 over a local or wide area telecommunications network.

Trend Micro, Inc.
EX1007-734

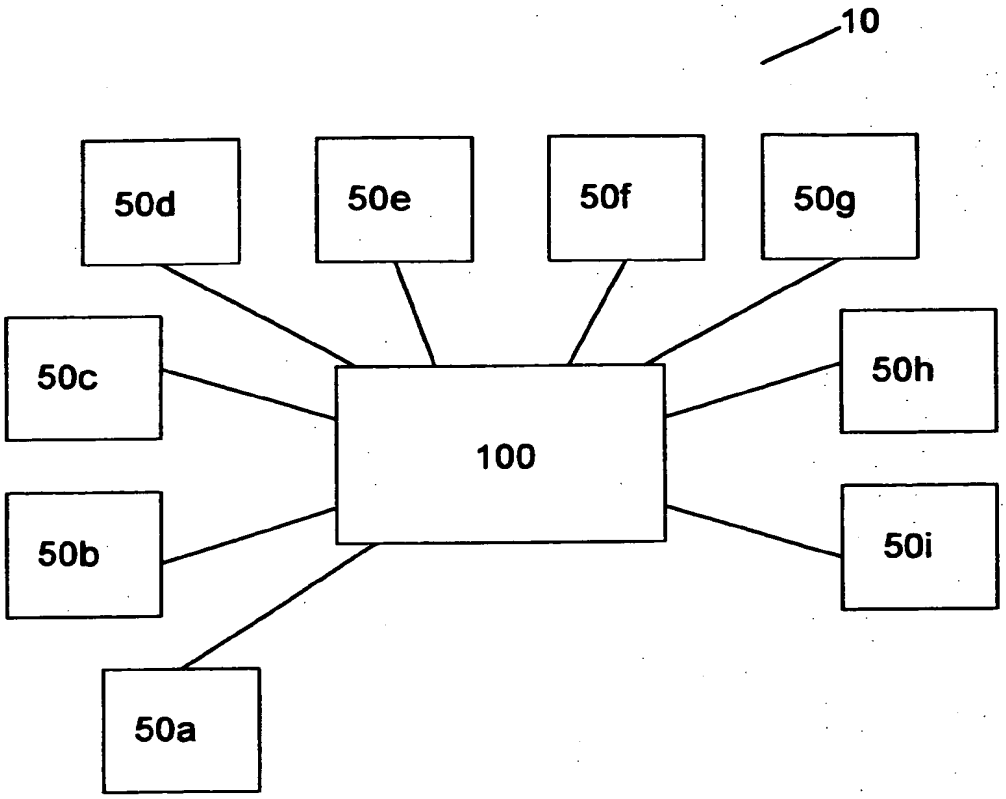


Fig. 1

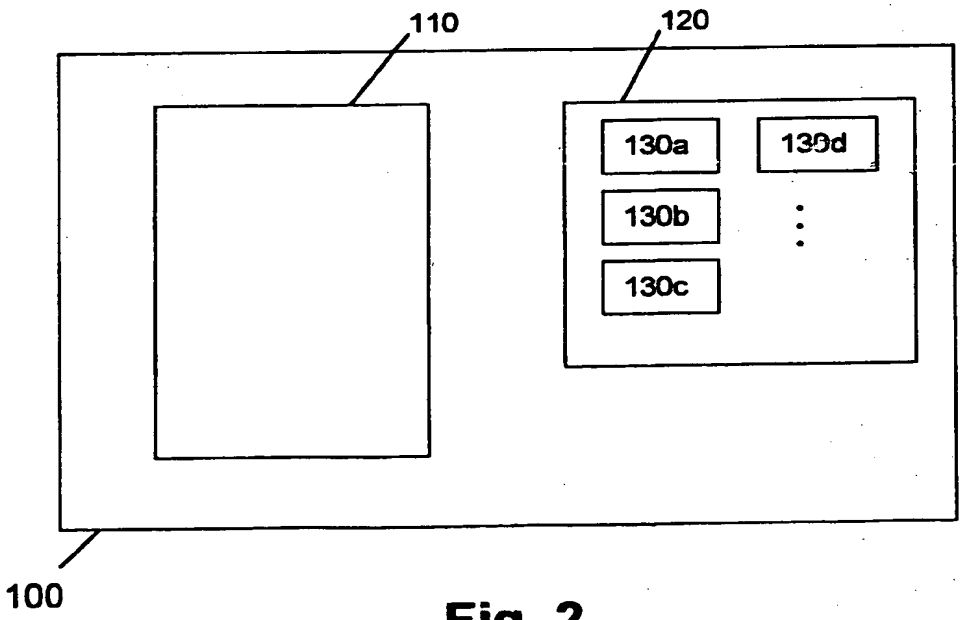


Fig. 2

Trend Micro, Inc.
EX1007-735

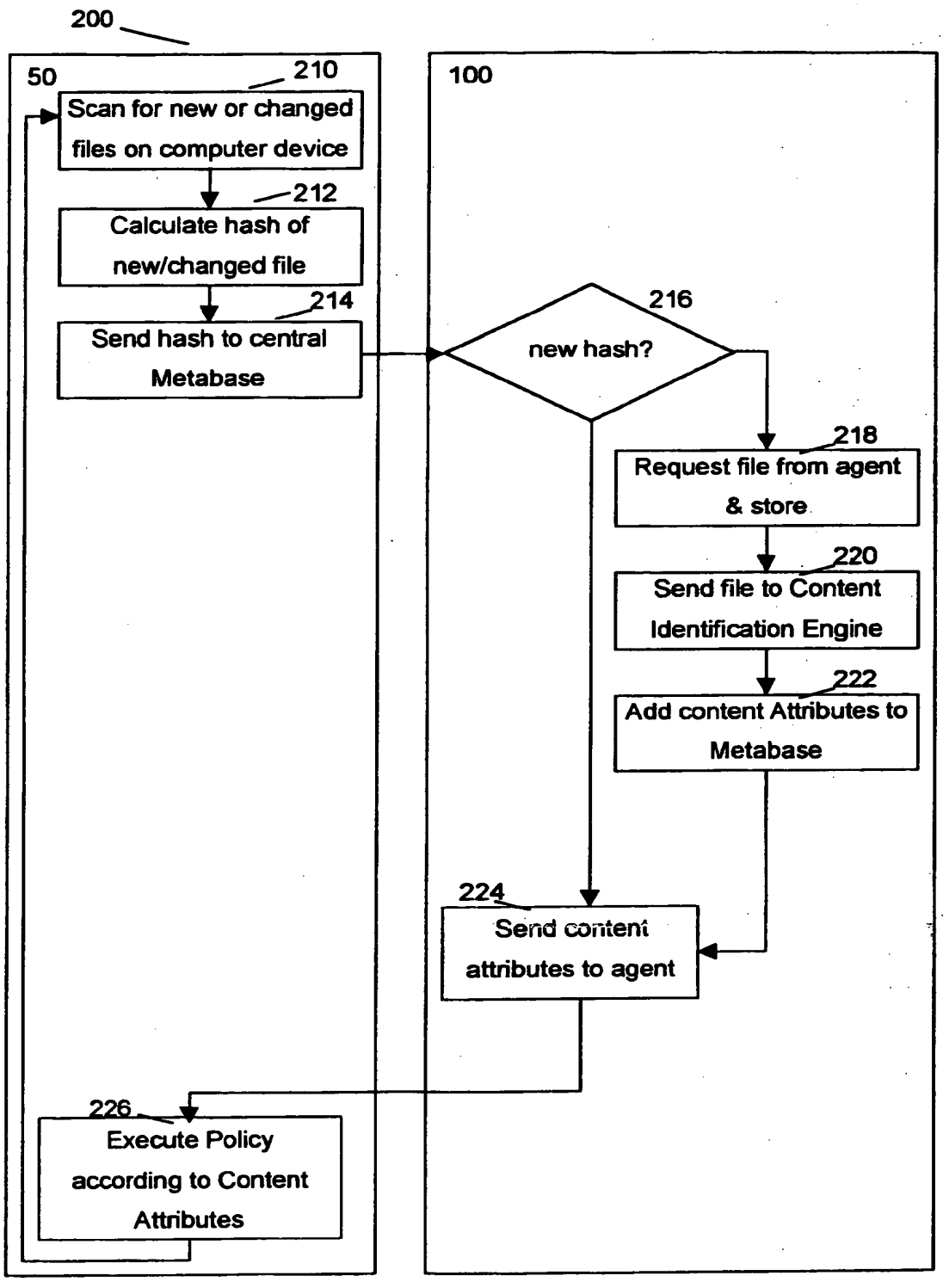


Fig. 3

Trend Micro, Inc.
EX1007-736

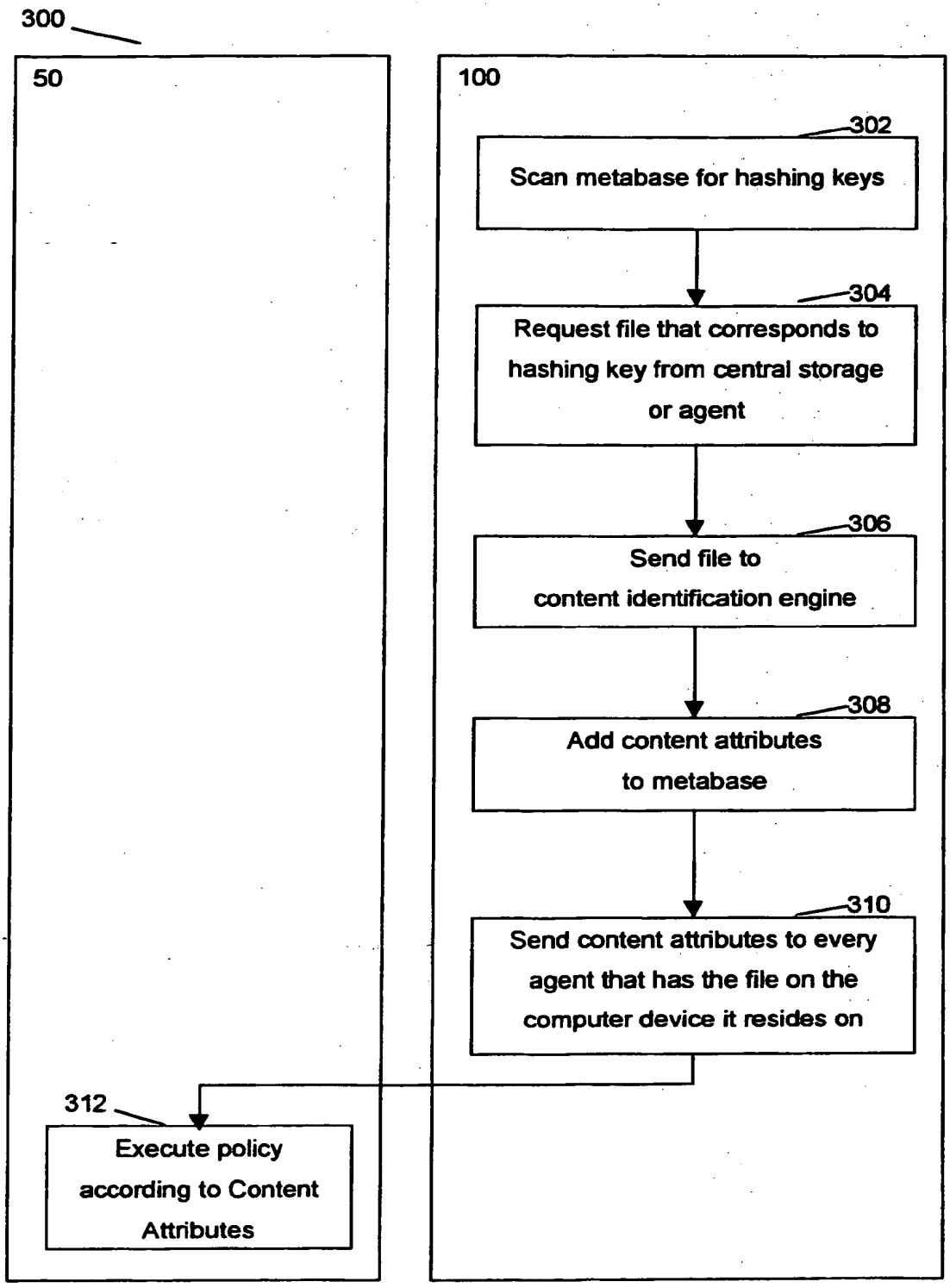


Fig. 4

Trend Micro, Inc.
EX1007-737

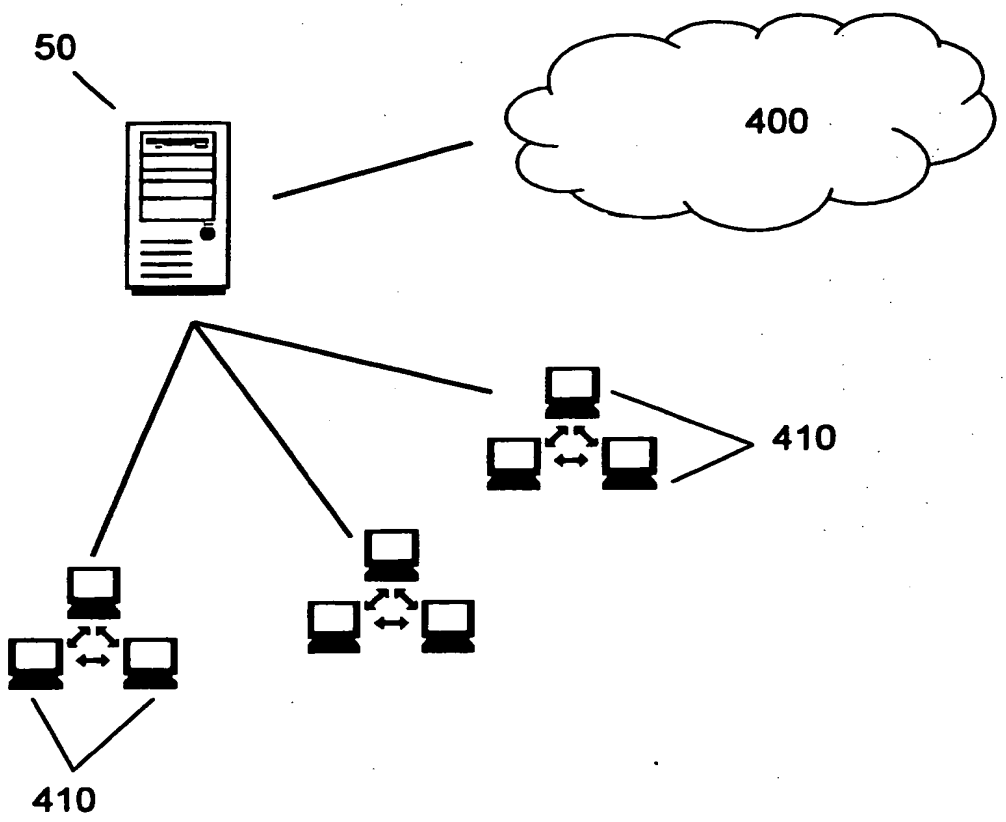


Fig. 5

Trend Micro, Inc.
EX1007-738

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 03 44 7310

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
X	EP 1 280 040 A (NETWORKS ASSOCIATES TECHNOLOGY) 29 January 2003 (2003-01-29) * the whole document *	1-9, 11-17, 20,21	H04L29/06 G06F1/00
A		10,18,19	

A	US 6 195 587 B1 (HRUSKA JAN ET AL) 27 February 2001 (2001-02-27) * column 1, line 4-9 * * column 1, line 50 - line 65 * * column 3, line 55 - column 5, line 67 * * column 1, line 50 - column 3, line 3 * * column 3, line 55 - column 5, line 40 *	1-6, 9-11, 13-20	

A	US 2003/120953 A1 (RADATTI PETER V) 26 June 2003 (2003-06-26) * paragraphs [0013]-[0015] * * paragraph [0045] * * paragraph [0047] * * paragraph [0051] *	1-4,7,8, 11, 13-15, 17,20	

A	EP 1 111 507 A (SECURE OYJ F) 27 June 2001 (2001-06-27) * paragraphs [0002]-[0004] * * paragraphs [0010]-[0012] * * paragraph [0015] * * paragraphs [0025],[0026] * * paragraph [0031] *	1,4-8, 12-16,20	

TECHNICAL FIELDS SEARCHED (Int.Cl.7)
H04L
G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	7 June 2004	Ruiz Sanchez, J

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

Trend Micro, Inc.
EX1007-739

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 03 44 7310

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

07-06-2004

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 1280040 A	29-01-2003	US 2003023866 A1	30-01-2003
		EP 1280040 A2	29-01-2003
US 6195587 B1	27-02-2001	GB 2283341 A	03-05-1995
		AU 8000394 A	22-05-1995
		DE 69407812 D1	12-02-1998
		DE 69407812 T2	16-07-1998
		EP 0725951 A1	14-08-1996
		WO 9512162 A1	04-05-1995
		JP 9504395 T	28-04-1997
US 2003120953 A1	26-06-2003	NONE	
EP 1111507 A	27-06-2001	GB 2353372 A	21-02-2001
		EP 1111507 A2	27-06-2001
		US 2001005889 A1	28-06-2001

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

Trend Micro, Inc.
EX1007-740

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☑ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
11 November 2004 (11.11.2004)

PCT

(10) International Publication Number
WO 2004/097602 A2

(51) International Patent Classification[7]: G06F 1/00

(21) International Application Number:
PCT/GB2004/001333

(22) International Filing Date: 29 March 2004 (29.03.2004)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
0309463.8 25 April 2003 (25.04.2003) GB

(71) Applicant *(for all designated States except US)*: **MESSAGELABS LIMITED** [GB/GB]; 1270 Lansdowne Court, Gloucester Business Park, Gloucester LG3 4AB (GB).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **SHIPP, Alexander** [GB/GB]; Star Internet, Brighouse Court, Barmwood, Gloucester GL4 3RT (GB).

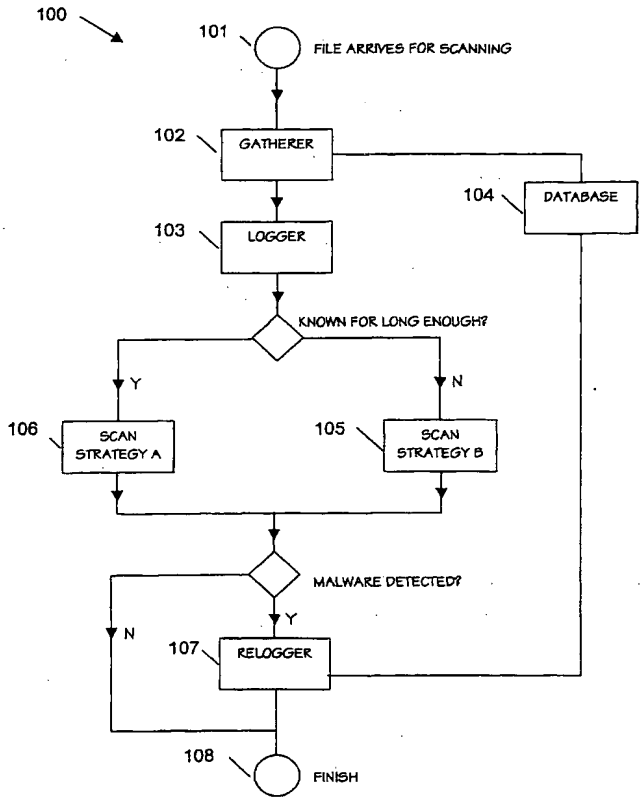
(74) Agents: **AYERS, Martyn** et al.; J.A. Kemp & Co., 14 South Square, Gray's Inn, London WC1R 5JJ (GB).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW),

[Continued on next page]

(54) Title: A METHOD OF, AND SYSTEM FOR, HEURISTICALLY DETERMINING THAT AN UNKNOWN FILE IS HARMLESS BY USING TRAFFIC HEURISTICS



(57) Abstract: A system for processing a computer file to determine whether it contains a virus or other maiware maintains a database of known files which it references to determine whether the file is an instance of a known file, and if so, whether it has been known about long enough that it can be regarded as safe. If it can be regarded as safe, the file is subject to less thorough processing for detecting malware, or no such processing at all.

BEST AVAILABLE COPY

WO 2004/097602 A2

Trend Micro, Inc.
EX1007-742

Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

A METHOD OF, AND SYSTEM FOR, HEURISTICALLY DETERMINING THAT AN UNKNOWN FILE IS HARMLESS BY USING TRAFFIC HEURISTICS

The present invention relates to a method of, and system for, heuristically determining that an unknown file is harmless by using traffic heuristics. This technique is especially applicable to situations where files enter a system, are checked, then leave, such as email gateways or web proxies. However, it is not intended to be limited to those situations.

Increasing use of the Internet, personal computers and local- and wide-area networks has made the problem of viruses and other malware (=malicious software) ever more acute.

There are numerous anti-virus packages available. These tend to be produced by specialist companies and are used by businesses and other organisations, home users, and by some internet service providers (ISPs) who scan e-mail and other network traffic on behalf of their customers as a value-added service. As new viruses and other malware arise, the package creators devise ways of detecting them and dealing with them and issue updates to their packages which customers can utilise. A common practice is to make the updates available for download over the internet, from the creator's website or ftp site.

Most anti-virus packages include a file-scanning engine and a database of characteristics of known viruses which are used by the scanning engine to determine whether a file being scanned is, or contains, a virus or other malware, or is likely to do so. The sort of update mentioned above typically includes an update to this database.

The scanning engine may implement a variety of heuristics to be applied, possibly selectively, to a file being scanned. Probably the most familiar kind of heuristic is signature detection, in which the file is examined for the occurrence of sequences or bytes, or patterns of such sequences, which are known to be characteristic of viruses in the package's virus database, though many other heuristics also exist, which can be used as well as or instead of signature detection.

The amount of malware in existence increases all the time, which makes the computational and storage resources necessary to detect it increasingly burdensome, particularly where the throughput of files is high, as is the case with ISPs.

According to the present invention, there is provided a system for processing a computer file to determine whether it contains a virus or other malware comprising:

a) means for generating data with regard to the file to characterise its identity and for thereby referencing a computer database to determine whether it is an instance of a known file;

b) means for selectively subjecting the file to a number of heuristic procedures to determine whether or not it contains, or is likely to contain, malware; and

c) means for determining, in dependence upon the record, if any, of the file in the database, whether the file can be regarded as safe and for controlling the means b) such that the file, if the file is to be regarded as safe, is either subject to less thorough processing than if it were not so regarded or not subject to processing by the means b) at all.

The invention also provides a method of processing a computer file to determine whether it contains a virus or other malware comprising:

a) generating data with regard to the file to characterize its identity and for thereby referencing a computer database to determine whether it is an instance of a known file;

b) selectively subjecting the file to a number of heuristic procedures to determine whether or not it contains, or is likely to contain, malware; and

c) determining, in dependence upon the record, if any, of the file in the database, whether the file can be regarded as safe and conducting the step b) such that the file, if the file is to be regarded as safe, is either subject to less thorough processing than if it were not so regarded or not subject to processing by the step b) at all.

The invention will be further described by way of non-limitative example with reference to the accompanying drawings, in which:

Figure 1 is a block diagram of a system embodying the present invention.

Figure 1 illustrates one form of a system 100 according to the present invention, which might be used, for example by an ISP as part of a larger anti-virus scanning system which employs additional scanning methods on files which are not filtered out as "safe" by the system of Figure 1. Files considered safe can if desired be subject to further processing to check for malware, but less intensively so than files not considered safe.

The rationale of the system 100 is that if a particular file has been scanned by a virus scanner, and found to be harmless the two possibilities exist: The file could really be harmless, or the file could contain something nasty which the virus scanner is as yet unable to detect.

As time goes by, the file (or another instance of it) may be scanned again, and still found to be harmless.

This time the file is more likely to really be harmless, rather than to be malware which the virus scanner is as yet unable to detect. This is because virus scanners are continually updated to detect new malware as the new malware is discovered. The longer the time that passes, the more likely it is that a suspicious person will submit a file containing malware to the developers of the scanner, who will analyse the file, and update their scanner to detect it.

As more and more instances of the file are scanned coming from different sources, then if these are all flagged as harmless, it becomes less and less likely the file is malware. This is because the more copies of a piece of malware exist, the more likely it is that somebody will become suspicious and submit a copy to scanner developers.

It is therefore possible to create a feedback engine which logs copies of files scanned, together with the source they originated from. The log is updated and examined as each file is scanned, and if files are found which have come from a sufficient number of sources, in sufficient quantities, and over a long enough period of time, then that file can be flagged as 'known about long enough'. This might mean that future copies are then not scanned further, or are scanned using less rigorous scans with fewer heuristics enabled, or are only scanned if the scanner has been updated since the last scan.

The system 100 operates according to the following algorithm:

1) A file arrives at an input 101 for scanning, perhaps as an email attachment, or a web download.

2) A 'gatherer' module 102 gathers information about the file, such as a checksum of the file contents and the source of the file (eg the IP address). The source may be passed through a one way trapdoor function, generating a hash, in order to preserve confidentiality. The information gathered is for comparison with information stored in a database 104 about known files so that it can be determined whether the file under consideration is an instance of a file recorded in database 104.

3) Based on the checksum derived by gatherer 102, a 'logger' module 103 updates the database 104 to indicate that one more instance of the file has been detected.

The logger 103 saves the current 'last seen' date as the 'previously scanned date', and then updates the 'last seen' date of the file's entry in the database 104. If this is the first instance of the file, the logger 103 also updates a 'first seen' date. If this is a new source, the logger 103 adds the source to a list, stored in database 104, of sources the file has originated from.

4) From the information stored (number of copies of the file seen, length of time file has been known about, number of sources) the logger 103 calculates whether the file has been 'known about long enough'. For this purpose, the logger 103 may assign a weighted score to each of these factors individually and then calculate an overall score by combining the weighted scores, e.g. by adding them up.

5) If the file has not been known about long enough, scan strategy B is undertaken at 105. This will be the most complete scan available.

6) If the file has been known about long enough, scan strategy A is undertaken at 106. This will be a less thorough scan than strategy B. This will be site-dependent as to how less thorough a scan is desired. At the extreme it might involve no scanning at all. It might involve scanning with fewer scanners; with heuristics not fully enabled or turned off; or (assuming the file has been seen at least once before) only with scanners that have been updated since the 'previously scanned date'

The scanning techniques available to the scanning strategies A and B may include any suitable heuristics, such as signature-based scanning, generating checksums from the file or selected regions if it, etc.

7) Following the scan strategy A or B, then if no malware was detected, processing stops at 108.

8) If malware was detected, then a 'relogger' module 107 is invoked. This clears out all database entries in database 104 which are associated with the file so that it cannot become 'known about long enough' in the future.

9) Processing of the current file finishes at 108, whereupon the system can retrieve the next file from a queue of files waiting to be processed.

CLAIMS

1. A system for processing a computer file to determine whether it contains a virus or other malware comprising:

a) means for generating data with regard to the file to characterise its identity and for thereby referencing a computer database to determine whether it is an instance of a known file;

b) means for selectively subjecting the file to a number of heuristic procedures to determine whether or not it contains, or is likely to contain, malware; and

c) means for determining, in dependence upon the record, if any, of the file in the database, whether the file can be regarded as safe and for controlling the means b) such that the file, if the file is to be regarded as safe, is either subject to less thorough processing than if it were not so regarded or not subject to processing by the means b) at all.

2. A system according to claim 1 wherein the controlling means c) controls the means b) in dependence on factors including the length of time for which the database indicates that the file has been known without malware-containing instances of it being detected.

3. A system according to claim 1 or 2 wherein the controlling means c) controls the means b) in dependence on factors including sources, recorded in the database, from which instances of the file have originated.

4. A system according to claim 1, 2 or 3 wherein the controlling means c) controls the means b) in dependence on factors including the number of times, recorded in the database, of instances of the file have been processed.

5. A system according to any one of the preceding claims, and including means for updating the database in dependence upon the result of the processing of the file by the means b).

6. A system according to claim 5 wherein the updating of the database, in the event of the means b) determining that the file contains, or is likely to contain, malware is such that the record thereof in the database is deleted, or updated so that it is no longer taken be safe.

7. A method of processing a computer file to determine whether it contains a virus or other malware comprising:

a) generating data with regard to the file to characterise its identity and for thereby referencing a computer database to determine whether it is an instance of a known file;

b) selectively subjecting the file to a number of heuristic procedures to determine whether or not it contains, or is likely to contain, malware; and

c) determining, in dependence upon the record, if any, of the file in the database, whether the file can be regarded as safe and conducting the step b) such that the file, if the file is to be regarded as safe, is either subject to less thorough processing than if it were not so regarded or not subject to processing by the step b) at all.

8. A method according to claim 7 wherein the determining step c) controls the step b) in dependence on factors including the length of time for which the database indicates that the file has been known without malware-containing instances of it being detected.

9. A method according to claim 7 or 8 wherein the determining step c) controls the step b) in dependence on factors including sources, recorded in the database, from which instances of the file have originated.

10. A method according to claim 7, 8 or 9 wherein the determining step c) controls the step b) in dependence on factors including the number of times, recorded in the database, instances of the file have been processed.

11. A method according to any one claims 7 to 10, and including the step of updating the database in dependence upon the result of the processing of the file by the step b).

12. A method according to claim 11 wherein the updating of the database, in the event of the step b) determining that the file contains, or is likely to contain, malware is such that the record thereof in the database is deleted, or updated so that it is no longer taken be safe.

13. A system for processing a computer file to determine whether it contains a virus or other malware substantially as hereinbefore described and with reference to the accompanying drawings

14. A method of processing a computer file to determine whether it contains a virus or other malware substantially as hereinbefore described and with reference to the accompanying drawings

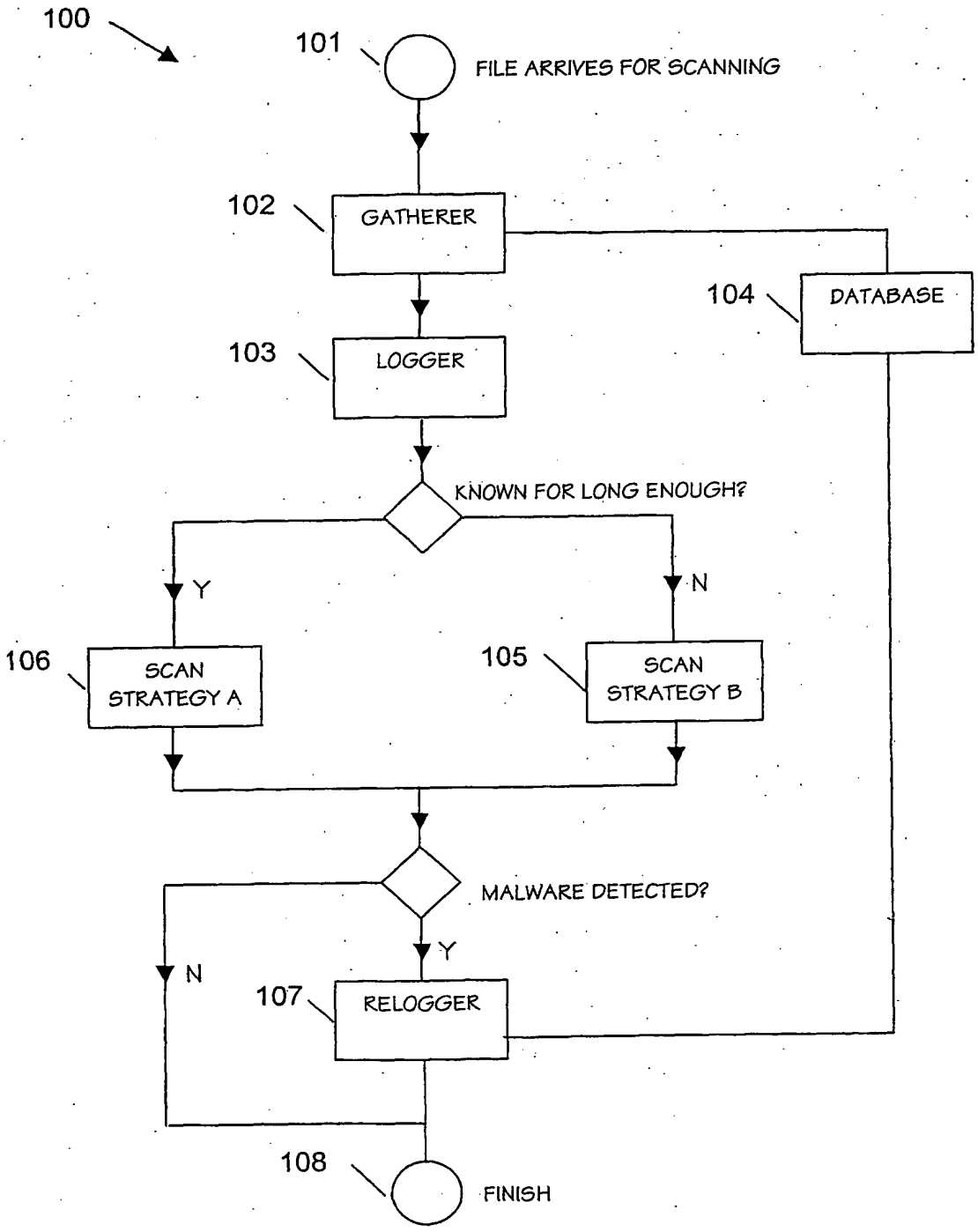

100
101
FILE ARRIVES FOR SCANNING
102
GATHERER
104
DATABASE
103
LOGGER
KNOWN FOR LONG ENOUGH?
Y
N
106
SCAN STRATEGY A
105
SCAN STRATEGY B
MALWARE DETECTED?
Y
N
107
RELOGGER
108
FINISH

Fig.1

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

OCT 31 2006
IAP44
PATENT & TRADEMARK OFFICE

Practitioner's Docket No.: 011765-0355986

PATENT

OIPE
IAP44
OCT 31 2006
PATENT & TRADEMARK OFFICE

Client Reference No.: AF/JG/P10717US

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: MELVYN MORRIS, ET AL.

Confirmation No: 4785

Application No.: 11/477,807

Group No.: 2131

Filed: June 30, 2006

Examiner: Unassigned

For: METHODS AND APPARATUS FOR DEALING WITH MALWARE

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SUBMISSION OF PRIORITY DOCUMENT

Attached please find the certified copy of the foreign application from which priority is claimed for this case:

Country	Application Number	Filing Date
Great Britain	0513375.6	06/30/2005

Date: October 31, 2006

P.O. Box 10500
McLean, VA 22102
Telephone: (703) 770-7900
Facsimile: (703) 770-7901
Customer Number: 00909

PILLSBURY WINTHROP SHAW PITTMAN LLP
Christine H. McCarthy
Registration No. 41844

Trend Micro, Inc.
EX1007-753

THIS PAGE BLANK (USPTO)

The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

I, [illegible] undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the [illegible] regulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Co[illegible]oller-General, hereby certify that annexed hereto is a true copy of the documents as orig[illegible]y filed in connection with the patent application identified therein.

I als[illegible]tify that the attached copy of the request for grant of a Patent (Form 1/77) bears an amen[illegible]nt, effected by this office, following a request by the applicant and agreed to by the Com[illegible]er-General.

In acc[illegible]ce with the Patents (Companies Re-registration) Rules 1982, if a company named in this ce[illegible]ate and any accompanying documents has re-registered under the Companies Act 1980 w[illegible]e same name as that with which it was registered immediately before re-registration save fo[illegible]ubstitution as, or inclusion as, the last part of the name of the words "public limited compan[illegible]heir equivalents in Welsh, references to the name of the company in this certificate and any [illegible]panying documents shall be treated as references to the name with which it is so re-registe[illegible]

In accorda[illegible]ith the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or P[illegible]

Re-registrat[illegible]der the Companies Act does not constitute a new legal entity but merely subjects the [illegible]ny to certain additional company law rules.

CERTIFIED COPY OF
PRIORITY DOCUMENT

Signed [signature]

Dated 12 July 2006

dti A DTI SERVICE

BEST AVAILABLE COPY

THIS PAGE BLANK (USPTO)

Patents Form 7/77

Patents Act 1977

(Rule 15)

The
Patent Office

THE PATENT OFFICE
C
30 JUN 2005
NEWPORT

Statement of inventorship and of right to grant of a patent

The Patent Office
Cardiff Road
Newport
South Wales
NP9 1RH

1. Your reference	**13532:JK**
2. Patent application number *(if you know it)*	0513375.6 — 30 JUN 2005
3. Full name of the or each applicant	**Retento Limited**
4. Title of the invention	**Computer Security**
5. State how the applicant(s) derived the right from the inventor(s) to be granted a patent	**By virtue of an agreement between the inventors and applicant**
6. How many, if any, additional Patents Forms 7/77 are attached to this form? *(see note (c))*	**0**

7. I/We believe that the person(s) named over the page *(and on any extra copies of this form)* is/are the inventor(s) of the invention which the above patent application relates to.

Signature: **Saunders & Dolleymore** Date: **29 June 2005**

8. Name and daytime telephone number of person to contact in the United Kingdom	**John Kensett** **01923 238311**

Notes

a) If you need help to fill in this form or you have any questions, please contact the Patent Office on 0645 500505.

b) Write your answers in capital letters using black ink or you may type them.

c) If there are more than three inventors, please write the names and addresses of the other inventors on the back of another Patents Form 7/77 and attach it to this form.

d) When an applicant does not declare any priority, or declares priority from an earlier UK application, you must provide enough copies of this form so that the Patent Office can send one to each inventor who is not an applicant.

e) Once you have filled in the form you must remember to sign and date it.

Trend Micro, Inc.
EX1007-757

THIS PAGE BLANK (USPTO)

Patents Form 7/77

Enter the full names, addresses and postcodes of the inventors in the boxes and underline the surnames

Melvyn Morris
Redmire Gap
Intake Lane
Turnditch
Derbyshire DE56 2LU
United Kingdom

Patents ADP number *(if you know it)*: 91271840001

Paul Stubbs
7 Rookery Road
Wyboston
Bedfordshire MK44 3AX
United Kingdom

Patents ADP number *(if you know it)*: 91271920001

Markus Hartwig
Exchange House
494 Midsummer Boulevard
Milton Keynes
Buckinghamshire MK9 2EA
United Kingdom

Patents ADP number *(if you know it)*: 91272000001

Reminder

Have you signed the form?

Patents Form 7/77

THIS PAGE BLANK (USPTO)

Patents Form 7/77

Enter the full names, addresses and postcodes of the inventors in the boxes and underline the surnames

Darren Harter
178 Barnwood Road
Hucclecote
Gloucester
Gloucestershire GL4 3JZ
United Kingdom

Patents ADP number *(if you know it)*: 91272800 1

Patents ADP number *(if you know it)*:

Patents ADP number *(if you know it)*:

Reminder

Have you signed the form?

Patents Form 7/77

THIS PAGE BLANK (USPTO)

Patents Form 1/77

Patents Act 1977

(Rule 16)

The **Patent Office**

J114381/ 002 D02396 P01/77FEE.....
02JUL05 30.00 CHEQUE 0513375.6

Request for grant of a patent
(An explanatory leaflet on how to fill in this form is available from the Patent Office)

THE PATENT OFFICE
C
30 JUN 2005
NEWPORT

0513375.6

The Patent Office

Cardiff Road
Newport
South Wales
NP10 8QQ 30 JUN 2005

1. Your reference:
(optional) **13532:JK**

2. Full name, address and postcode of the applicant or of each applicant *(underline all surnames)*:
Retento Limited
3 New Park Place
Pride Park
Derby DE24 8DZ
United Kingdom

Patents ADP number *(if you know it)*: 9127234001

If the applicant is a corporate body, give the country/state of its incorporation: **United Kingdom**

3. Title of the invention: **Computer Security**

4. Name of your agent *(if you have one)*:
Saunders & Dolleymore BECK GREENER

"Address for service" in the United Kingdom to which all correspondence should be sent *(including the postcode)*:
9 Rickmansworth Road FULWOOD HOUSE
Watford 12 FULWOOD PLACE
Hertfordshire WD18 0JU LONDON WC1V 6HR

Patents ADP number *(if you know it)*: **1453001**

see sheet 30/6/05

5. Priority declaration: Are you claiming priority from one or more earlier-filed patent applications? If so, please give details of the application(s):

Country	Application number *(if you know it)*	Date *(day/month/year)*

6. Divisionals, etc: Is this application a divisional application, or being made following resolution of an entitlement dispute about an earlier application? If so, please give the application number and filing date of the earlier application:

Number of earlier UK application	Date of filing *(day/month/year)*

7. Inventorship: (Inventors must be individuals not companies) (Please tick the appropriate boxes)

Are all the applicants named above also inventors? YES ☐ NO ☑

If yes, are there any other inventors? YES ☐ NO ☐

8. Are you paying the application fee with this form? YES ☑ NO ☐

Trend Micro, Inc.
EX1007-763

THIS PAGE BLANK (USPTO)

9. Accompanying documents: not counting duplicates, please enter the number of pages of each item accompanying this form:

Continuation sheets of this form:

Description: 26

Claim(s): 9

Abstract: 1

Drawing(s): 3 + 3 RM

If you are not filing a description, please give details of the previous application you are going to rely upon:

Country	Application number	Date of filing

10. If you are also filing any of the following, state how many against each item.

Priority documents:

Statement of inventorship and right to grant of a patent (Patents Form 7/77): 1

Request for search (Patens Form 9A/77): 1

Request for substantive examination (Patents Form 10/77):

Any other documents: (please specify)

11. I/We request the grant of a patent on the basis of this application.

Saunders & Dolleymores

Signature(s): Date: **29 June 2005**

12. Name, e-mail address, telephone, Fax and/or mobile number, if any of a contact point for the applicant:

John Kensett
01923 238311
info@dolleymores.com

Warning

After an application for a patent has been filed, the Comptroller of the Patent Office will consider whether publication or communication of the invention should be prohibited or restricted under section 22 of the Patents Act 1977. You will be informed if it is necessary to prohibit or restrict your invention in this way. Furthermore, if you are resident in the United Kingdom and your application contains information which relates to military technology, or would be prejudicial to national security or the safety of the public, section 23 of the Patents Act 1977 prohibits you from applying for a patent abroad without first getting written permission from the Patent Office unless an application has been filed at least 6 weeks beforehand in the United Kingdom for a patent for the same invention and either no direction prohibiting publication or communication has been given, or any such direction has been revoked.

Notes

A leaflet on how to fill in this form is available from the Patent Office. If you would like a copy of the leaflet it is available on our website at http://www.patent.gov.uk/patent/info/fact05.pdf or alternatively you could telephone 08459 500505 or send an email to enquiries@patent/go.uk to request a copy.

THIS PAGE BLANK (USPTO)

DUPLICATE

Computer Security

This invention relates to computer security. In particular, it relates to methods and systems for detecting whether a process operated on a computer is likely to be one which is safe to operate or is likely to be malware.

Background to the Invention

The term "malware" is intended to mean any process which might possibly be harmful to the computer, its software, firmware or data stored on the computer or its performance, reliability, control and utilisation. Non-limiting examples of malware are viruses, Trojans, worms, spyware, adware or similar.

Anti-malware systems have been in existence for nearly as long as malware processes have existed and several service providers produce anti-virus and other anti-malware software and services. Generally, the method of dealing with malware is that when new types of malware are released upon the Internet for example these are eventually detected and possibly infect user's computers or special monitoring computers specially connected to the Internet for this purposes. Once new items of malware have been detected, then the service providers in the field generate signatures which attempt to deal with these and these signatures are then released as updates to their anti-malware programs. Heuristic methods have also been employed.

These systems work well for protecting against known risks. However, since they rely on signature files being generated and/or updated, there is inevitably a delay between a

new piece of malware coming into existence or being released and the signatures for combating that malware being generated/updated and supplied to users. Thus, users are at risk from brand new malware for a certain period of time which might be up to a week or even more. During this time, many users can inadvertently become infected before proper signatures have been disseminated.

WO 2004/0976 A2 describes a system which analyses computer files received or generated by a local computer and compares these with a database to determine whether the file is known and if so whether it has been known about long enough that it can be regarded as safe. If it can be regarded as safe, then the file is subject to a less thorough processing for detecting malware then a program about which it is not known whether it is safe or not.

The system disclosed therein appears to only look at whole files rather than processes. A process may be included as part of a file or may be a separate item altogether. Furthermore, this previously proposed system assumes that once a file has been known about for "long enough" it is inherently safe. This is not necessarily the case in real life when, for example, the active payload of a virus or Trojan may only be programmed to happen at a particular date or event which may be many months or even years after the process has been first run or is released. Thus, just looking at the age of a file is an unsatisfactory way of determining whether it is properly safe and will remain so.

Furthermore malware may be programmed to become effective based on other criteria which may not be date or duration based, for example threshold values or other parameters.

An object of the invention is to be able to detect whether a process or component operated on a computer is likely to be one which is safe to operate or is likely to be malware.

A further object is to seek to eliminate or reduce risk in malware in the inter-regnum between its emergence and the generation of signatures. This might be known as zero-date risk.

Brief Summary of the Invention

According to the present invention in a first aspect, there is provided a computer network comprising at least one local terminal, at least one anti-malware service provider database terminal remote from the terminal, a local database on the local terminal which is populated as processes are run or downloaded onto the terminal, a server database on the or each server database terminal and means for determining, when a process is found at the local terminal, whether that process is known or unknown at the local terminal and is considered safe or unsafe and, if unknown, whether the process is known at the server database and is considered safe or unsafe, and means for dynamically altering the database contents and safe/unsafe status using predetermined conditions.

The invention further provides a method of protecting a local computer terminal against malware, comprising providing a local process signature database on the local computer and a remote database on a computer which is remote from the local terminal, searching the local database at the terminal when a process is run to determine whether the process is known or not at that terminal and, if known, is considered safe or unsafe; if the process is not known then searching the remote database to determine if the process is

known and, if so, is determined safe or unsafe; and if the process is known at the remote database submitting the safe/ unsafe result back to the local terminal.

According to a further aspect, there is provided a computer system including means for minimising the risk of malware infection, the system comprising at least one local terminal and a community database comprising a plurality of databases each on a different computer remotely distributed from each other and from the local terminal, whereby signature and/or mask files representative of known processes are stored and built up on the community database to obtain, over time, a community database storing signature and/or mask files relevant to or known processes and indicating whether each process is considered safe or unsafe.

According to a yet further aspect there is provided a method of determining the protection which a computer terminal has from malware, the method comprising providing a database remote from the terminal; transmitting from the terminal to a database details of all or selected security products operating at a point in time on that terminal; obtaining similar information from other terminals connected to the database; searching for the same combination of security products used by machine; and identifying any malware processes which were not identified by said other machine having that particular combination of security products and process.

In a further aspect, the invention comprises a database populated with information from a plurality of client terminals representative of the combination of security products and processes used by each client terminal at a point in time, and details of any known

malware processes that were not detected by said unique combination for each particular terminal combination.

Embodiments of the invention may respond to events, not to recognition of files themselves, and to time, both points in time when events occur and to elapsed time, and may also be responsive to volume and acceleration of apparently associated events captured by the database.

Description of the Drawings

Embodiments of the invention will now be described, by way of example only, with reference to the accompanying drawings, in which:

Figure 1 shows a network;

Figure 2 is a flowchart showing the running of a process;

Figure 3 is a flowchart showing further processing steps; and

Figure 4 is a flowchart showing an information obtaining stage.

Detailed Description of Preferred Embodiments of the Invention

Referring to Figure 1, a computer network is generally shown as being based around a distributed network such as the Internet 1. A local computer terminal 2 is connected to the Internet by any suitable means as will be known in the art. The terminal 2 may be a computer, a PDA, mobile phone, an interactive television or any other device capable of loading and operating processes. A process may be a computer file, may be part of a file or may be a sub-program, macro, web page or any other piece of code to be operated by the terminal or any other event whether executed, emulated, simulated or interpreted. A

process 3 is shown in the figure and this may be applied directly to the terminal locally, as shown by line 4 or may be downloaded to the terminal via the Internet as shown by line 5.

Systems according to the invention utilise a database connected to the Internet and which can interact with the local terminal when that terminal runs a process to determine whether the process is safe or unsafe. The database may be provided on a single server connected to the Internet but is preferably provided over a distributed "community" database 5 as shown in Figure 1, which comprises a number of data servers 5a, 5b, ... 5n. The provision of a distributed community database of this type firstly enables each individual device to require less storage and more importantly makes the database much less vulnerable to denial of service or other attacks since if one server is attacked, then, particularly if entries in the database are duplicated, performance of the database will not be significantly affected.

The community database is populated, over time, with information relating to each process run on all of the local terminals within the system. A user of a terminal might subscribe to the services of the database provider or may alternatively be able to download software for free, which enables him to use such services. Data representative of each process may take the form of a so-called signature or key relating to the process and its affects and to a mask which sets out the parameters of the processes performance and operation. The mask will be described in more detail further below.

Figure 2 shows how the system may run in one embodiment.

At the start point 21, a computer process is run at the local terminal, either directly or being downloaded or run direct from a site over the Internet 1. The local terminal then searches its local database 6 at step 22 to search for a signature or key representing that particular process. If the local signature is present, it will indicate either that the process is considered to be safe or OK (in the manner described further below), or will indicate that that process is considered unsafe. An unsafe process might be one which has been found to be malware or to have unforeseen or known unsafe or malevolent results arising from its running. If the signature indicates that the process is OK, then that process is allowed to run at step 23. If the signature indicates that the process is not OK, then the process is stopped 24.

Alternatively, the user may be presented with a choice of whether to allow the process to run or not, but is informed that the process is not considered safe. Ultimately, it will be the user's choice whether to risk running the process or not.

In preferred embodiments there may be more than two 'states': GOOD-BAD-UNKNOWN AT LOCAL TERMINAL-UNKNOWN IN 'COMMUNITY DATABASE'. If the process is GOOD then it is, for example, always allowed to run. If BAD then it is NOT allowed to run - the user is given no choice in this. If UNKNOWN locally or in the Community then the user may be given informed choices of viewing or not. Other states and levels may of course be used.

If the process is unknown, then this means that the process has not been run before at that local terminal. In this case, details of the process are passed over the Internet or

other network to the community database and the community database is searched for a signature for that process at step 25. The community database is supplied with signatures representative of programs or processes run by each monitored local terminal (and in a typical version in the field there may be several millions of these terminals and so any processes which are newly released upon the Internet, etc will soon be found and signatures created). When the community database is searched, if the signature is found and indicates that that process is safe, then firstly a copy of the signature is sent to the local terminal database (step 26), to begin populating the local database so that the local terminal has this information immediately to hand then next time the process is run and a message is passed back to the local terminal allowing the process to run.

If the signature is found and this indicates for some reason that the process is unsafe, then again the signature is copied back to the local database and marked UNSAFE (step 27), and then the process is either stopped (or not allowed to run) or the user given an informed choice whether to run it or not.

If after the entire community database has been searched the process is still unknown, then it is assumed that this is a brand new process which has never been seen before in the field. A signature is therefore created representative of the process at step 28 and this signature is initially marked as bad or unsafe at step 29. The signature is copied to the local database of the terminal which first ran the process at step 30 and, in addition, a copy of the file or process itself may be requested at step 31 by the community database from the local terminal. A message may then be passed to the local terminal which

instructs the terminal not to run the process or alternatively the user may be given informed consent as to whether to allow the process to run or not.

If the user chooses to run a process which is considered unsafe because it is too new, then that process may be monitored by the local terminal and/or community database and, if no ill effect occurs or is exhibited, it may then be considered to be safe after a period of time of n days for example. Alternatively, the community database may keep a log of each instance of the process which is found by the many local terminals forming part of the network and after a particular number of instances have been recorded, possibly with another particular number of instances or the process being allowed to run and running safely, the signature may then be marked as safe rather than unsafe. Many other variations of monitoring safety may be done within this concept.

The signature of a process may comprise or include a mask which can be built up with use of that process and which indicates the particular types of behaviour to be expected from the process. If a process is allowed to run on a local terminal, even if the initial signature search 22 indicates that the process is safe, then operation of that process may be monitored within the parameters of the mask. The mask might indicate for example, the expected behaviour of the process; any external requests or Internet connections that that process might legitimately have to make or call upon the local terminal to make, including details of any ports or interfaces which might be required to be opened to allow such communication; any databases, either local or over a local area network or wide area network or Internet, which may be expected to be interrogated by that

process and so on. Thus, the mask can give an overall picture of the expected "normal" behaviour of that signature of that process.

The behaviour of the process is continually monitored both at the local terminal and also by continually sending information to and from the community database relating to that process to determine whether the process is running within its expected mask. Any behaviour which extends beyond the mask is identified and can be used to continually assess whether the process continues to be safe or not. Thus, if the process, on a regular basis (say monthly or yearly) opens a new port to update itself for example or to obtain regular data, then this information is flagged. If it is found that the process has done this on other terminals and has had no ill effects, or this behaviour is known from other processes and known as being safe, then this behaviour might be considered as safe behaviour but the mask is then modified to allow for this. If it has been found previously that this new behaviour in fact causes unsafe or malevolent results, then the process can then be marked as unsafe even if previously it was considered safe. Similarly, if the process attempts to connect to a known unsafe website, database or to take action which is known as generally being action only taken by unsafe programs then again the process may be considered to be unsafe.

Thus, embodiments of the invention can continually modify a processes safe or unsafe rating with time.

This is shown in Figure 3. However, Figure 3 also introduces the concept that any process can be pre-authorised. Typically, it will be pre-authorised by a trusted partner, such

as a major software company, a verification authority, a Government department and so on. Pre-authorisation enables a supplier of a new process, which has not been released before, to get pre-authorisation for that process, and includes the provision by that supplier of a mask detailing the expected and allowable behaviour of that process. Thus, the mask defines the limits of that processes behaviour. Pre-authorisation may be done by the software provider themselves (if they are considered to be a Trusted Provider), or by the software provider providing details of that process to a trusted authority such as a Government or verification authority who in turn verifies the process and then enables the relevant electronic "documentation" to be provided to the database provider for its required pre-authorisation technique. So a new process or software item by a trusted software provider can be pre-authorised and have no risk of being prevented from running. Similarly, if a bespoke item of software is created for an organisation then that organisation may be able to give or obtain immediate pre-authorisation

Referring to Figure 3, when a process is run, the local and/or community database are searched as before at step 31. If the process is not a pre-authorised one, then the steps of Figure 2 are taken, for example and the process might be allowed to run or not at step 32. If the process is pre-authorised, as determined by step 33, then it is immediately allowed to run, step 34. This may be the end of the matter. However, in a preferred variation, the process is then monitored whilst running, and is monitored each time it is run in the future in a monitoring state step 35 to determine whether its behaviour falls within its pre-authorised mask 36. If the behaviour falls within the pre-authorised behaviour then the process is allowed to continue to run. If the behaviour extends beyond the allowed mask,

such as by trying to instigate further processes or connections which have not been pre-authorised, then this behaviour is flagged at an alert step 37.

Various actions could be taken at this stage. The process might simply not be allowed to run. Alternatively, the trusted authority which initially enabled pre-authorisation might be contacted, who may be able to confirm that this behaviour is acceptable or not. If it is acceptable, then the mask could be modified. If not acceptable, then the process might be marked as unsafe. Many other actions may be taken upon the noting of such an alert state.

If the process has been found not to be pre-authorised at step 33 and is allowed to run, then the process is monitored at step 38 in order to generate a mask 39 representative of the normal behaviour of that process. Data representative of this mask might be sent to the community database for scanning when other terminals run that process. By continually monitoring a process each time it is run or during running of the process, any behaviour which differs from previous behaviour of the process can be noted and the mask can modified, or this behaviour might be used to determine that a process which was once considered safe is now considered to be unsafe.

When a user of a local terminal subscribes to a service according to an aspect of the invention, or downloads software enabling this, his system may periodically provide information to the community database representative of all or selected ones of the software products loaded on or available to his system. In particular, this may be all the locally-loaded security products (such as anti-malware systems including anti-virus software, anti-

spyware, anti-adware and so on), firewall products, etc and includes version details, specific settings and details of which signature files are currently loaded, and also information such as which files are operating and the particular version and software settings at any time and similar types of software. Whilst these may be provided individually to the community database, it may prefer to generate a signature or key at the local terminal representative of the sum information (eg by a hashing or compression function) which can then be transmitted to the database for storage. Figure 4 shows how the details of all local security products, versions, signature files, firewall settings, etc 40 are used to create a key 41. The key is transmitted to the community database. Since the community database is provided with such information from many, possibly millions, of user's terminals, it can search for other terminals having the same combination of security products including the same setting, signature files loaded and so on.

The community database can also provided with details of processes run by every machine and knows whether a process has been detected by each local machine or not, hence the database can be used to obtain information as to whether a particular, specific, combination of security products, including settings and signature files existing at a point in time, render a particular terminal susceptible or vulnerable to any particular malware process. That is, if the database knows that a terminal in the past has version A of anti-virus product B, with downloaded signature update C and also has a firewall D with particular settings E and perhaps anti-spyware software F with signature updates G, but that a particular malware process P was not detected by this combination of programs at that point in time, then this information can be provided to a local terminal which is known as

having that combination of security programs/settings and can be used to indicate that that terminal is vulnerable in the short term to attack by that particular malware process.

This information can be presented to the user either by displaying a window 43 on the screen display of the local terminal or by directing the user to a particular website which explains the position in more detail. In other words, the community database, when provided with information relating to all the security products on a particular terminal at a particular time, searches for events for processes marked as "bad" that occurred on systems with that particular mix of security products and were not locally detected. It might indicate that process P1 was "bad" and process P2 was "bad" and so on. This information is then fed back to the terminal user either directly or by directing him to a website. A message might be provided on a window on his terminal which reads, for example;

"Your system currently comprises the following security products, versions and signature files:

Anti-virus system A, version A_1, signatures files A_{11} dated 01/06/05,

Anti-spyware system B, version B_1, signature files B_{11} dated 04/06/05

Firewall C, updated 30/05/05, etc ...

Be aware that your system may not protect against malware process P1, P2, etc"

The message will generally give these bad processes the names commonly associated with them (eg Sobig virus, and so on). The user may be directed to further guidance on a website for example where he may be given suggestions of how to deal with this situation.

While there may be a suggestion to download different software, it could alternatively be suggestions for downloading particular patches, closing particular ports, or so on. This information could be provided virtually in real-time and will generally only be applicable to a particular point in time.

It will be seen from the above that the community database can store a great deal of information about the security and other software products installed on a user's terminal and also about potential risks and malware processes and their effects. Up to now, anti-virus software providers generally provide databases on their website which a user can look up for details of particular viruses and possible suggestions as to preventing infection by them or eradicating any infection. A database generated in accordance with embodiments of the present invention, can provide a lot more detail including:

the name of malware and all the other names it is known by;

the way it propagates its prevalence; and

which folders and authorised processes it infects or affects, and so a more extensive database than before can be built up which users can interrogate. The database can also include details, as described above, of the interaction between these or various security products on their terminal and various malware products in order for the user to determine himself the best combination. Also, by continually obtaining details of the software set up on user's terminals and storing these as archive files, if a user's terminal does crash or is affected by malware, the contents of the archive of the community database may be used to provide a more effective backup to restore the user's terminal to a previous safe configuration.

Note that whereas much prior art Anti-Virus software attempts to identify by a scan of files newly introduced to the local computer or network that are known to be bad by virtue of a match against an index file of malware signatures and stops any part of said files from executing, embodiments of the present invention trap the execution activity of all code components on the subject computer and check the signature of each code component against a local database and/or remote database of previously benign activity. If any component is about to execute a not previously known type of activity, that execution is, by default, blocked. Accordingly, in embodiments of the invention all executing code is monitored and any code executing in a manner not previously known to be benign is blocked. Sometimes it is not possible to trap which particular component (of a known suite of executing code components) is producing suspect activity. In such a case composite signatures can be used to tie back or relate to a particular suspect event and thereby to a specific component that is suspect malware. For example, the well known browser "Internet Explorer" has different installations (for different operating systems, configurations etc.) which use many different types of subsidiary code components. At execution time it is not always possible to distinguish which particular code component is triggering a previously unknown event. The community database can collect data that allows a signature to be made of any user's configuration of composite sets of components, to achieve this.

In some embodiments of the invention meta data can be captured in the community database, this enables current and refined newly devised heuristics to be run on the community meta database. This in turn allows the ability to model heuristic techniques which themselves can be propagated to the agent level when the new heuristic has been

proved. "what if" modelling can also be done on the community metadata – e.g. what if we mark this code component as bad for the time being, or, what other applications among what number of users might be affected?

Amongst other types, the types of meta data captured might be:

"alias" – related to the frequency with which a code component appears in code that has a name different from other named code it has already been seen to be present in, and;

"event key" – being a signature that ties an event to a particular configuration of component code element present on the machine on which the event has occurred. This permits lowest common denominator/ higher-common factor type analysis on the community metadata to determine code components that are malware.

It should be noted that in contrast to previously proposed "signature" based methods, with methods according to the present invention there is intrinsic protection even for offline users since the system can be arranged, while out of contact with the community database (i.e. off-line or because of a problem connection therewith), to "lock down" (e.g. prevent execution of, or give the user informed consent) an unknown process. In effect, if a process is unknown at the local terminal then in such circumstances it can be considered, at least initially, automatically to be bad or suspect.

A further situation arises when users wish to install authorised software while offline.

By prompting the user when they attempt to install new software while offline, the execution of the installation can itself be "authorised" – the authorisation by the user is treated as a "Genesis" event and will be so termed hereinafter. However there are some processes commonly used in installation of software that communicate with other existing programs on the installation machine, e.g. Microsoft's MSIEXEC.

The Genesis approach involves a process that generates signatures as a result of the user's authorisation on the user's terminal.

Further, systems according to the invention may download signatures of so-called "licensed installer", being known code components, to users machines while they are online. This allows the operation of a method such that a "licensed installer" is permitted to execute while a Genesis event is "current", i.e. within a period of minutes after an authorisation from the user. There may be refinements to this method, such as to prevent licensed installers executing if a media change has occurred during the currency of a Genesis event.

The method may be such that a "licensed installer" together with any child processes of the licensed installer is permitted to execute while a GENESIS event is "current", i.e. within a period of minutes after an authorisation from the user. However, any unknown processor, which may be malware, may still be detected and processed.

There may be refinements to this method, such as to prevent licensed installers executing if a media change has occurred during the currency of a Genesis event and super-trusted installers such as for example 'Windows Update', where all new processes created by the super-trusted installer are marked immediately as safe.

Preferably signatures of licensed installers are ALWAYS downloaded, as and when added to the community database, to an agent machine while online. There probably are not that many of these.

When the terminal is connected to a network, the other aspect to this feature is that the system will, if the about to be installed program is not known to the community database, block the installation, or alert the user, i.e. "you are about to install some software, this software is not known to [community]. Are you sure you wish to proceed?". Having a limited number of licensed installers facilitates download, as minimal data is required, compared to downloading virus signature files.

Note that embodiments of the invention can be applied within servers (e.g. in a server-enterprise environment) and therefore can be executed on any computer or terminal within a network, including ones which may be connected to a network or not, from time to time. Even a server itself may be a local 'terminal' as the term is used in this specification.

The potentially high volume of active users (i.e. terminals) (which will in practice be many millions) gives a high probability of at least one of them being infected by new malware. The speed of propagation is detected and recorded in embodiments of the

invention so that the propagation of malware can be detected and malware designated as bad, and can be used to identify an incident emerging across the community as malware, or simple volume of infection can be used as a trigger.

In a further feature, difference of naming, of an otherwise identical piece of code combined with acceleration of first attempts to execute the code within the community allows pattern matching that will show up an identical signatured piece of code as bad.

This feature allows the statement in some embodiments, "nothing will propagate in our community faster than X without being locked down", so that if any process or event propagates quicker over a given duration, it is marked as bad.

This process can be automated by the identification of the vector of propagation in the community, from timestamp data held in the community database and the marking of a piece of code that has these attributes as bad. By comparison, it is believed that all other anti-malware providers rely on a simplistic known bad model and therefore are reliant primarily on malware infection actually occurring on terminals and being reported.

One method for plotting the acceleration of a suspect piece of code involves timestamping of each report of the same piece of code, showing how many terminals had reported the code within successive timeslots, within seconds or minutes initially but then perhaps against a more logarithmic set of lengthening time periods i.e. the method by which timestamping is done usually cannot be maintained at the same rate as during the earliest period of first detection. This logarithmic method avoids the database becoming swamped

with stamp data by avoiding a linear method of data capture, e.g. every few seconds of all incidents of the same code. Where appropriate, in some embodiments, suspect files can be uploaded automatically from individual terminals to the community database for more detailed examination.

Amongst other features and inventions, the following aspects form part of the present invention, any one or more of which, or any combination of which, or any one or more feature or combination hereinbefore described, are considered as possible claimable inventions in their own right.

a) The concept of a known and trusted parent process having an identifying component in its signature that uniquely defines the process (referred to here as a "gene"). By virtue of such gene being inherited in the signature of all of the parent's child processes and through them further child processes (referred to here as "issue") such signatures are all automatically recorded in the local database as good. The concept also allows the inherited gene to be removed should any issue abuse it, such as by creating a suspect child or one that becomes suspect. This concept, combined with Genesis, allows the extension of the trusted status of parent processes notwithstanding the involvement of surrogate programs (like Microsoft's MSIExed for example) in generating further issue. The marking of child process signatures as good might otherwise fail because of the involvement of surrogates like MSIExec, and other licensed installer programs. Alternatively, a parent process which is not necessarily known and trusted may have a unique identifier, which may possibly be considered inherently safe, unsafe or given other attributes. More than one identifier (gene) may be used. The attribute may be security related or may be related to other aspects such

as being subscription related so that a process can only be executed whilst a subscription is current, and so on.

b) Parental creation of a signature for a child or children including the ability for these to be automatically marked as either good or bad depending on the parent's behaviour and determination. Note that in some embodiments the product can 'watch' or monitor the birth of a child process and automatically create the signature upon arrival.

c) Genesis creation through a licensed or trusted installer or surrogate who may pass on the true parents (99% good) genes. The 'genesis' process allows the product to capture the birth and apply the appropriate genetic determination.

d) A feature may be included that allows for automatic forfeiture of a child's inherited ability to trigger the automatic creation of signatures on any further births because the child, as parent has already produced bad offspring. Preferably, a rule is that if a file has one bad offspring then the inherited ability can be automatically removed.

e) An ability to watch the clones or identical twins (i.e. the same process running on other systems in the community) to compare the pattern of their off-spring to ensure that Genesis events are definitely part of the same genetic line and remove any determination for an off spring which is not prevalent as a child of the clones or identical twins, or to make the determination unknown causing it to be watched intensely throughout the community before a determination can be made.

Some of the features a) to e) represent a solution to the issues of producing a product which can be used effectively without 100% reliance on being permanently connected to the internet, which is often impractical. Examples of this are Windows Update, and other processes used more and more by vendors who wish to be able to roll out other product updates automatically across the worldwide web.

Possible benefits of these types of features above conventional software are as follows. Antivirus software tends to have a cache of known bad signatures. The problem is keeping this up to date. Take the simple example of someone buying a new computer. The computer comes with an antivirus product preloaded with a signature cache. Between the time when the PC was built, shipped to the store and bought by the user several days or weeks will have passed. The user starts the PC and is exposed to any new virus or malware which was created after the PC was built. Full protection requires the user to connect to the internet and download updates. This cannot be guaranteed to occur ahead of other activities by the user on the internet (almost physically impossible to guarantee). With a local cache of known good processes, as in the embodiments of the present invention, it is possible to ship the computer/terminal preloaded with a pre-generated cache of signatures for all of the good (clean) software preloaded by the computer manufacturer. In this case the user can connect to the internet knowing that any new or updated programs will be immediately detected and verified. Also any auto-updating software can function forcing signatures to be automatically built for its children and more remote off-spring (i.e. grandchildren, great-grandchildren, etc).

f) The ability to monitor a process along with any optional sub-processes as an homogenous entity and then compare the activities of the top level process throughout the

community and deduce that certain, potentially malevolent practices only occur when one or more specific sub-processes are also loaded. This allows effective monitoring (without unnecessary blocking) of programs like Internet Explore (browsers) whose functionality may be easily altered by downloadable optional code that users acquire from the internet, the principal source of malevolent code today.

g) The ability, using the community database to make early diagnosis, or simply to take precautionary measures to stop potentially fast propagating worms and malware very, very early in their life cycle. Given that it is theoretically possible to create a worm which could infect every computer connected to the internet within a matter of a few minutes this feature is highly desirable.

h) The ability to make even faster determination by combining data defining the speed of propagation of a new piece of software with meta data collected by the Agent and fed to the database. This includes monitoring processes that attempt to conceal themselves from the user by randomly changing name and/or location on the local terminal, or every computer or network on which it installs. It also includes a process's attempt to create an identical copy (i.e. with identical code contents) on the computer but with a different name. This is a classic attribute of a worm.

i) The use of meta data derived from terminal agents defining the behaviour of a process across the community and held in a central database to model, test and create new automated rules for use in the central database and as rules that may be added to those held

and used in the local (terminal) database to identify and determine the agent's (system) response to new or unknown processes and process activity.

j) The use of two signature processes. One which defines the entire files and will change with almost any change to the files content. The other which attempts to define only the processing instructions of the process which changes much less. By tracking the occurrences of both signatures individually appearing with different counterparts it is possible to identify processes which have been changed or have been created from a common point but which have been edited to perform new, possibly malevolent functionality.

k) The use of an event mask which enables an agent to monitor a known process and detect if it performs abnormal events which could indicate the emergence of a new purpose from a Trojan. This enables the agent to take immediate protect action to block the abnormal activity and to report it to the central database.

l) A technique to enable an efficient and practical form of communication between the agent (i.e. local terminal) and the remote database which allows the local terminal and community to operate without unnecessary delay or impact on the users' enjoyment of the computer and processes he/she choose to use.

m) The technique of recording which security products, operating system, updates, settings and security patches are installed and active when an event occurs and to use this to help assess the likehood of a malevolent process bypassing specific security

products or combinations thereof. To use this information to warn users about the need to bolster their defences to prevent attack.

Claims

1. A computer network comprising at least one local terminal, at least one anti-malware service provider database terminal remote from the terminal, a local database on the local terminal which is populated as processes are run or downloaded onto the terminal, a server database on the or each server database terminal and means for determining, when a process is found at the local terminal, whether that process is known or unknown at the local terminal and is considered safe or unsafe and, if unknown, whether the process is known at the server database and is considered safe or unsafe, and means for dynamically altering the database contents and safe/unsafe status using predetermined conditions.

2. A system as claimed in Claim 1, wherein the server database is distributed over a plurality of server computers.

3. A method of protecting a local terminal against malware, comprising providing a local process signature database on the local and a remote database on a computer which is remote from the local terminal, searching the local database at the terminal when a process is about to be run to determine whether the process is known or not at that terminal and, if known, is considered safe or unsafe; if the process is not known then searching the remote database to determine if the process is known and, if so, is determined safe or unsafe; and if the process is known at the remote database submitting the safe/ unsafe result back to the local terminal.

4. A method as claimed in Claim 3, wherein when a process is first seen, a signature file is created at the local terminal and/or remote database representative of that process.

5. A method as claimed in Claim 3 or Claim 4, wherein if the process is not known at the remote database, a signature of the process is created at the remote database and initially marked as unsafe and a copy of the signature is sent to the local database.

6. A method as claimed in Claim 3, wherein if a process is marked as safe by the local or remote database it is allowed to run.

7. A process as claimed in Claim 3, wherein if a process is marked as unsafe by the local or remote database it is not allowed to run.

8. A process as claimed in Claim 3, wherein if a process is marked as unsafe by the local or remote database the user of the local terminal is provided with an informed choice as to whether to allow the process to run or not.

9. A method as claimed in Claim 3, wherein a mask is generated for each or a collection of processes, which mask defines the acceptable behaviour for that process or collection.

10. A method as claimed in Claim 3, wherein the mask includes one or more parameter selected from the group comprising;

the expected behaviour of the process or collection of processes; any external requests or Internet connections that that process or collection might legitimately have to make or call upon the local terminal to make, including details of any ports or interfaces which might be required to be opened to allow such communication; any databases, either local or over a local area network or wide area network or Internet, which may be expected to be interrogated by that process or collection.

11. A method as claimed in Claim 9, wherein after the process is allowed to run, its behaviour is monitored and if its behaviour extends beyond that allowed by the mask an alert is raised.

12. A method as claimed in Claim 10, wherein after the alert is raised, a determination is made as to whether the modified behaviour renders the process safe or unsafe, even if the process was originally considered safe.

13. A method as claimed in Claim1 or Claim 3, wherein a process may be pre-authorised.

14. A method as claimed in Claim 13, wherein the process is pre-authorised by a software provider or by a trusted authority.

15. A method as claimed in Claim 13, wherein if a process is not pre-authorised and is allowed to run, its behaviour is monitored and is used to create a mask representative of its behaviour, the behaviour continuing to be monitored to determine whether it expands

beyond the mask, in which case the modified behaviour can be analysed to determine whether the mask itself should be modified to take the modified behaviour into account or whether the modified behaviour indicates that acceptable behaviour has been exceeded and the process should now be considered unsafe.

16. A computer system including means for minimising the risk of malware infection, the system comprising at least one local terminal and a community database comprising a plurality of databases each on a different computer remotely distributed from each other and from the local terminal, whereby signature and/or mask files representative of known processes are stored and built up on the community database to obtain, over time, a community database storing signature and/or mask files relevant to or known processes and indicating whether each process is considered safe or unsafe.

17. A system as claimed in Claim 16, wherein even after signatures and masks have been established, behaviour of a process is monitored to determine whether its status is still safe or unsafe.

18. A method of determining the protection which a computer terminal has from malware, the method comprising providing a database remote from the terminal; transmitting from the terminal to a database details of all or selected security products operating at a point in time on that terminal; obtaining similar information from other terminals connected to the database; searching for the same combination of security products used by machine; and identifying any malware processes which were not identified by said other machine having that particular combination of security products and

process.

19. A method as claimed in Claim 18, further comprising providing information to the user of said first local terminal that his system may be susceptible to attack by said malware processes.

20. A method as claimed in Claim 19, wherein the combination of processes includes security products, versions, signature files loaded and other security products.

21. A method as claimed in Claim 19, wherein information is provided to a user on a display associated with the local terminal.

22. A method as claimed in Claim 19, wherein the information is displayed on a website.

23. A database populated with information from a plurality of client terminals representative of the combination of security products and processes used by each client terminal at a point in time, and details of any known malware processes that were not detected by said unique combination for each particular terminal combination.

24. A method as claimed in claim 3 whereas files, or data representative of files can be uploaded from the local database to the remote database.

25. A method as claimed in claim 3 wherein composite sets of data are analysed to

determine their status.

26. A method as claimed in claim 3 wherein metadata is collected and stored in the local and/or remote database.

27. A method as claimed in claim 26 when the metadata includes at least one member of the set comprising "alias" and "Event Key".

28. A method as claimed in claim 3 whereas if a local terminal is not connected to the remote database and a process unknown to that terminal is attempted to be executed an alert is raised that the process might be unsafe.

29. A method as claimed in claim 28 when the possibly unsafe process is not allowed to execute.

30. A method as claimed in claim 3 wherein a process unknown to a terminal can be preauthorised for loading onto or executing on said terminal while the terminal is not connected to the remote database.

31. A method as claimed in claim 30 wherein one or more "licensed installer" signatures are pre-supplied to said terminal.

32. A method as claimed in claim 3 wherein volume of infection and/or speed of propagation is used to determine whether a process is safe or unsafe.

33. A method as claimed in claim 32 wherein a vector of propagation is used to determine whether a process is safe or unsafe.

34. A method as claimed in claim 32 or 33 wherein propagation is determined by timestamping each report of the same process, and providing an indication of the number of terminals reporting attempted invocation of the process within successive time slots.

35. A method as claimed in class 34 wherein an at least partially logarithmic timescale is used.

36. An anti-malware system, arranged to detect the spread of new processes across a network and to consider a process as bad if the volume or speed of propagation exceeds a predetermined threshold.

37. An anti-malware system as claim in claim 36 wherein each report of the same process is timestamped and comprising counting the number of terminals in the network reporting that process in successive time periods.

38. An anti-malware system as claimed in claim 37 wherein a logarithmic set of time periods is used.

39. A method as claimed in claim 3 wherein there are more than two possible states representing different levels of safe or unsafe, and the process is allowed to be run or not, or the user given informed choice, depending on the level.

40. A method as claimed in claim 39 wherein the states are GOOD-BAD-UNKNOWN LOCALLY and UNKNOWN IN REMOTE DATABASE.

41. A method as claimed in claim 3 wherein an application may be labelled as super trusted and any child processor generated, or instigated as a result thereof, be considered safe at least for a period.

42. A computer program product or method, wherein a signature relating to a parent process includes a component that uniquely identifies it and which is inherited in the signature of any child or other descendant of the parent, such unique identifier being used to ascribe one or more attributes to the descendants.

43. A product or method as claimed in Claim 42, wherein the parent process is known and trusted and whereby an attribute is that the descendant process is safe.

44. A product or method as claimed in Claim 42 or 43, wherein if the parent or a descendant exhibits behaviour which might be unsafe or suspicious the attribute is, or is changed to, the process being unsafe.

45. A product or method as claimed in any of Claims 42 to 44, wherein the attribute is security related.

46. A product or method as claimed in any of Claims 42 to 45, wherein the attribute is not security related.

Abstract

Computer Security

A computer network is disclosed with minimised risk of malware infection, comprising at least one local terminal, at least one malware service provider database remote from the terminal, a local database on the local terminal which is allowed to populate and processes are run or downloaded onto the terminal, a server database on the or each server database terminal and means for determining, when a process is found at the local terminal, whether that process is known or unknown at the local terminal and is considered safe or unsafe and, if unknown, whether the process is known at the server database and is considered safe or unsafe, and means for dynamically altering the database contents and safe/unsafe status using predetermined conditions.

OCT 31 2006
PATENT & TRADEMARK OFFICE

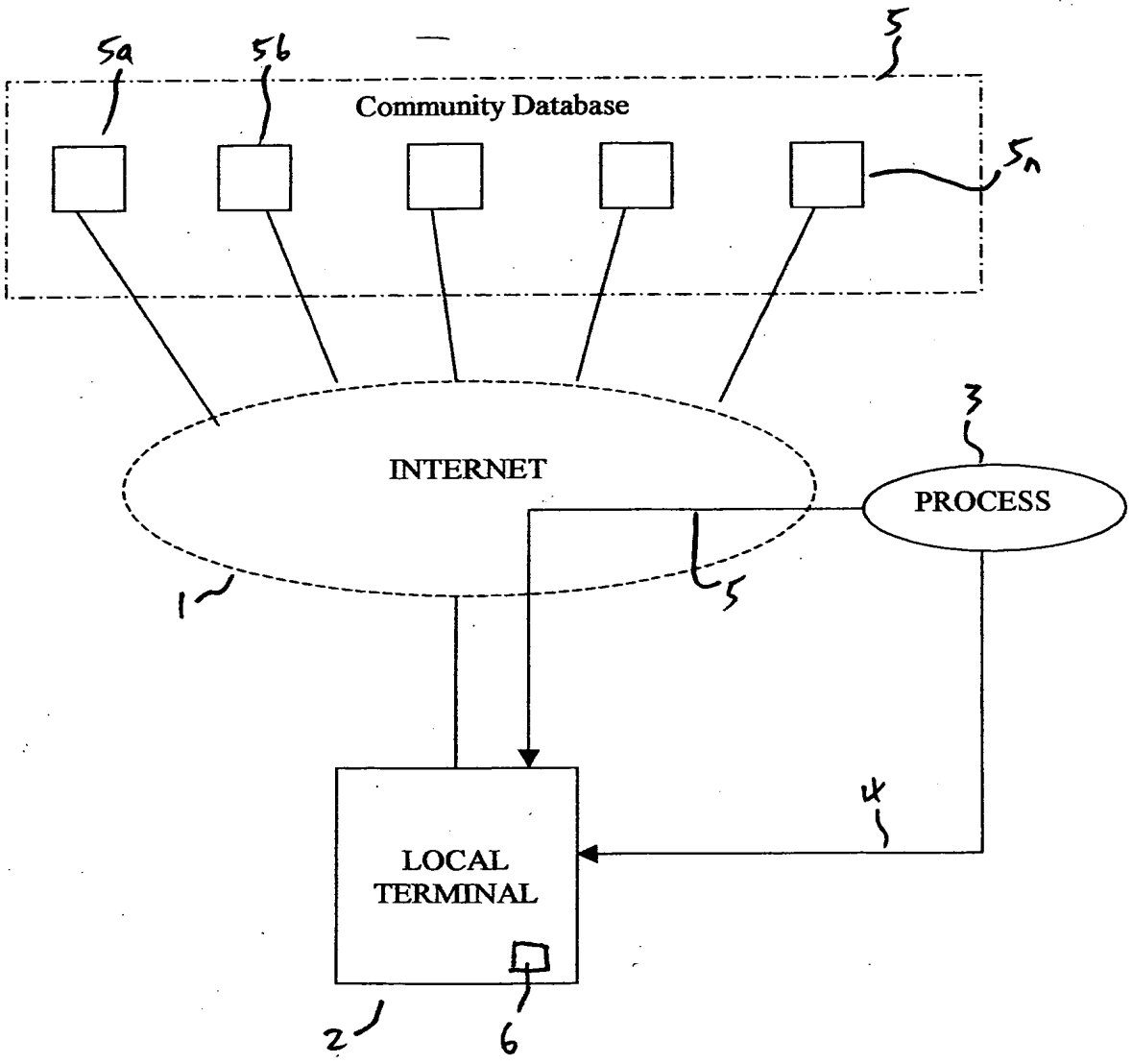


FIG 1

THIS PAGE BLANK (USPTO)

NOT TO BE AMENDED

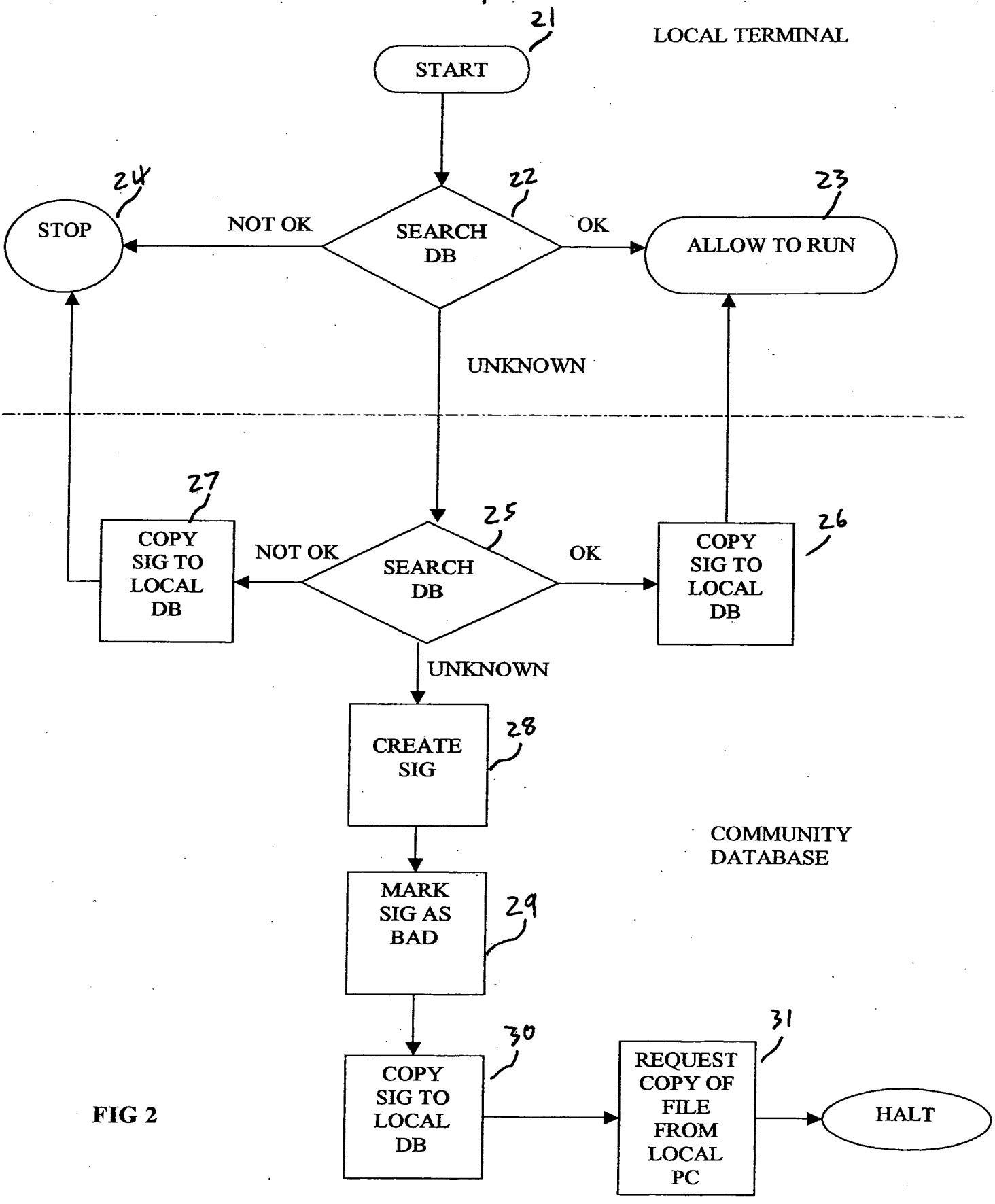


FIG 2

Trend Micro, Inc.
EX1007-805

THIS PAGE BLANK (USPTO)

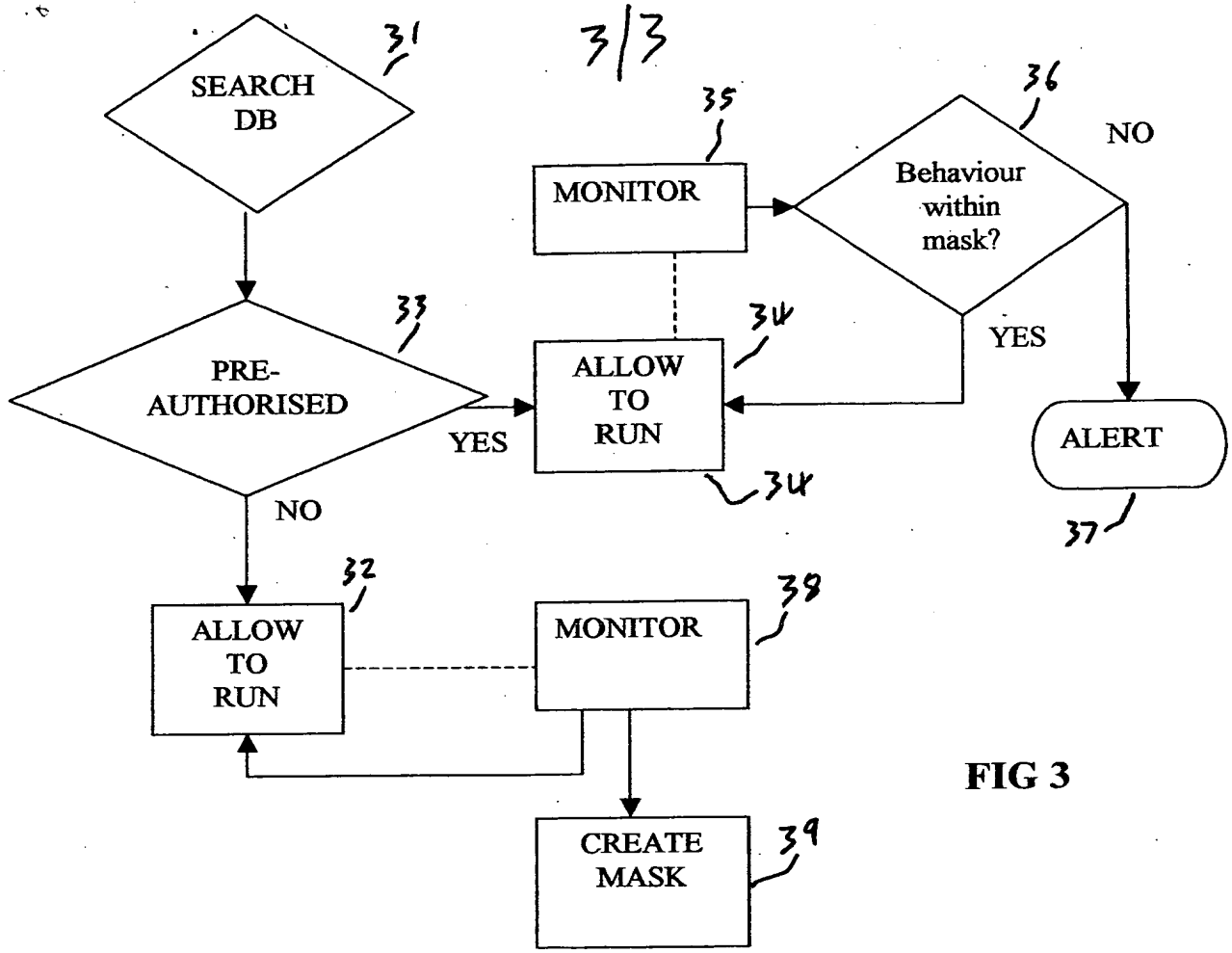

SEARCH DB
31
MONITOR
35
Behaviour within mask?
36
NO
YES
PRE-AUTHORISED
33
YES
ALLOW TO RUN
34
ALERT
37
NO
ALLOW TO RUN
32
MONITOR
38
CREATE MASK
39

FIG 3

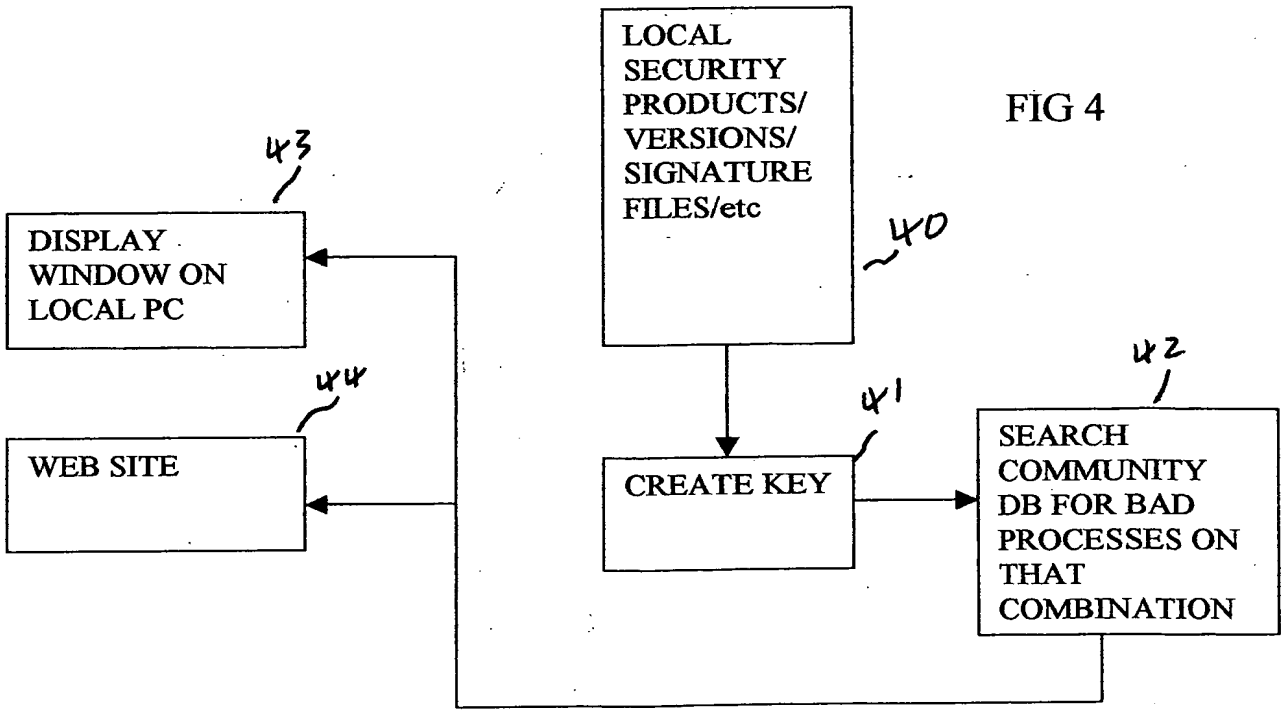

LOCAL SECURITY PRODUCTS/ VERSIONS/ SIGNATURE FILES/etc
40
43
DISPLAY WINDOW ON LOCAL PC
44
WEB SITE
41
CREATE KEY
42
SEARCH COMMUNITY DB FOR BAD PROCESSES ON THAT COMBINATION

FIG 4

THIS PAGE BLANK (USPTO)

OIPE IAP56
SEP 26 2006
PATENT & TRADEMARK OFFICE

Attorney Docket: 011765-0355986
Client Reference: AF/JG/P10717US

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re PATENT APPLICATION of: MELVYN MORRIS ET AL.
Application No.: 11/477,807 Confirmation No: 4785

Filed: June 30, 2006 Group Art Unit: 2131

Examiner: Unassigned

Title: METHODS AND APPARATUS FOR DEALING WITH MALWARE

RESPONSE TO NOTICE TO FILE MISSING PARTS

Mail Stop Missing Parts
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

In response to the Notice to File Missing Parts of Application mailed July 27, 2006, a copy of which is enclosed, Applicants herewith submit the following:

- Executed Combined Declaration and Power of Attorney of Inventors
- Assignment and Recordation Cover Sheet

Please see attached PTO/SB/17 for payment of fees.

Respectfully submitted,
PILLSBURY WINTHROP SHAW PITTMAN LLP

By _____
Christine H. McCarthy
Registration No. 41844
Telephone: (703) 770.7743

Date: September 26, 2006
P.O. Box 10500
McLean, VA 22102
Customer No.: 00909

Trend Micro, Inc.
EX1007-809

SEP 26 2006
PATENT & TRADEMARK OFFICE
IAP56

UNITED STATES PATENT AND TRADEMARK OFFICE
UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371 (c) DATE	FIRST NAMED APPLICANT	ATTORNEY DOCKET NUMBER
11/477,807	06/30/2006	Melvyn Morris	011765-0355986

CONFIRMATION NO. 4785

00909
PILLSBURY WINTHROP SHAW PITTMAN, LLP
P.O. BOX 10500
MCLEAN, VA 22102

FORMALITIES LETTER

Date Mailed: 07/27/2006

NOTICE TO FILE MISSING PARTS OF NONPROVISIONAL APPLICATION

FILED UNDER 37 CFR 1.53(b)

Filing Date Granted

Items Required To Avoid Abandonment:

An application number and filing date have been accorded to this application. The item(s) indicated below, however, are missing. Applicant is given **TWO MONTHS** from the date of this Notice within which to file all required items and pay any fees required below to avoid abandonment. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a).

- The statutory basic filing fee is missing.
  Applicant must submit $ **300** *to complete the basic filing fee for a non-small entity. If appropriate, applicant may make a written assertion of entitlement to small entity status and pay the small entity filing fee (37 CFR 1.27).*
- The oath or declaration is missing. *A properly signed oath or declaration in compliance with 37 CFR 1.63, identifying the application by the above Application Number and Filing Date, is required.*
  Note: If a petition under 37 CFR 1.47 is being filed, an oath or declaration in compliance with 37 CFR 1.63 signed by all available joint inventors, or if no inventor is available by a party with sufficient proprietary interest, is required.

The applicant needs to satisfy supplemental fees problems indicated below.

The required item(s) identified below must be timely submitted to avoid abandonment:

- Additional claim fees of **$2350** as a non-small entity, including any required multiple dependent claim fee, are required. Applicant must submit the additional claim fees or cancel the additional claims for which fees are due.
- To avoid abandonment, a surcharge (for late submission of filing fee, search fee, examination fee or oath or declaration) as set forth in 37 CFR 1.16(f) of $130 for a non-small entity, must be submitted with the missing items identified in this letter.

09/27/2006 CCHAU1 00000111 033975 11477807

01 FC:1011 300.00 DA
02 FC:1111 500.00 DA
03 FC:1311 200.00 DA
04 FC:1051 130.00 DA
05 FC:1202 1350.00 DA
06 FC:1201 1000.00 DA

SUMMARY OF FEES DUE:

Total additional fee(s) required for this application is **$3480** for a Large Entity

- **$300** Statutory basic filing fee.
- **$130** Surcharge.

- The application search fee has not been paid. Applicant must submit **$500** to complete the search fee.
- The application examination fee has not been paid. Applicant must submit **$200** to complete the examination fee for a large entity

- Total additional claim fee(s) for this application is **$2350**

  - **$1000** for **5** independent claims over 3.
  - **$1350** for **27** total claims over 20.

Replies should be mailed to: Mail Stop Missing Parts
Commissioner for Patents
P.O. Box 1450
Alexandria VA 22313-1450

A copy of this notice MUST be returned with the reply.

Office of Initial Patent Examination (571) 272-4000, or 1-800-PTO-9199, or 1-800-972-6382
PART 1 - ATTORNEY/APPLICANT COPY

OIPE IAP56 SEP 26 2006 PATENT & TRADEMARK OFFICE

PTO/SB/17 (01-06)
Approved for use through 07/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

Fees pursuant to the Consolidated Appropriations Act, 2005 (H.R. 4818).

FEE TRANSMITTAL
For FY 2006

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT ($) 3,520.00

Complete if Known	
Application Number	11/477,807
Filing Date	June 30, 2006
First Named Inventor	MELVYN MORRIS
Examiner Name	
Art Unit	2131
Attorney Docket No.	011765-0355986

METHOD OF PAYMENT (check all that apply)

☐ Check ☐ Credit Card ☐ Money Order ☐ None ☐ Other (please identify): _____

☒ Deposit Account Deposit Account Number: 033975 Deposit Account Name: PILLSBURY WINTHROP SHAW PITTMAN LLP

For the above-identified deposit account, the Director is hereby authorized to: (check all that apply)

☒ Charge fee(s) indicated below ☐ Charge fee(s) indicated below, **except for the filing fee**

☒ Charge any additional fee(s) or underpayments of fee(s) under 37 CFR 1.16 and 1.17 ☒ Credit any overpayments

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

FEE CALCULATION (All the fees below are due upon filing or may be subject to a surcharge.)

1. BASIC FILING, SEARCH, AND EXAMINATION FEES

	FILING FEES		SEARCH FEES		EXAMINATION FEES		
		Small Entity		Small Entity		Small Entity	
Application Type	Fee ($)	Fee ($)	Fee ($)	Fee ($)	Fee ($)	Fee ($)	Fees Paid ($)
Utility	300	150	500	250	200	100	1,000.00
Design	200	100	100	50	130	65	
Plant	200	100	300	150	160	80	
Reissue	300	150	500	250	600	300	
Provisional	200	100	0	0	0	0	

2. EXCESS CLAIM FEES

Fee Description	Fee ($)	Small Entity Fee ($)
Each claim over 20 (including Reissues)	50	25
Each independent claim over 3 (including Reissues)	200	100
Multiple dependent claims	360	180

Total Claims		Extra Claims	Fee ($)		Fee Paid ($)
47	- 20 or HP =	27	x 50.00	=	1,350.00

HP = highest number of total claims paid for, if greater than 20.

Indep. Claims		Extra Claims	Fee ($)		Fee Paid ($)
8	- 3 or HP =	5	x 200.00	=	1,000.00

HP = highest number of independent claims paid for, if greater than 3.

Multiple Dependent Claims	
Fee ($)	Fee Paid ($)

3. APPLICATION SIZE FEE

If the specification and drawings exceed 100 sheets of paper (excluding electronically filed sequence or computer listings under 37 CFR 1.52(e)), the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).

Total Sheets		Extra Sheets		Number of each additional 50 or fraction thereof	Fee ($)		Fee Paid ($)
61	- 100 =	-39	/ 50 =	_____ (round up to a whole number)	x 250.00	=	0.00

4. OTHER FEE(S)

	Fees Paid ($)
NLate Declaration Surchage	130.00
Other (e.g., late filing surcharge): Assignment Recordation Fee	40.00

SUBMITTED BY

Signature		Registration No. (Attorney/Agent) 41844	Telephone 703.770.7743
Name (Print/Type)	Christine McCarthy		Date September 26, 2006

This collection of information is required by 37 CFR 1.136. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 30 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

OIPE IAP56
SEP 26 2006
PATENT & TRADEMARK OFFICE

Attorney Docket: 011765-0355986
Client Reference: AF/JG/P10717US

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re PATENT APPLICATION of: MELVYN MORRIS ET AL.
Application No.: 11/477,807 Confirmation No: 4785

Filed: June 30, 2006 Group Art Unit: 2131

Examiner: Unassigned

Title: METHODS AND APPARATUS FOR DEALING WITH MALWARE

RESPONSE TO NOTICE TO FILE MISSING PARTS

Mail Stop Missing Parts
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

In response to the Notice to File Missing Parts of Application mailed July 27, 2006, a copy of which is enclosed, Applicants herewith submit the following:

- Executed Combined Declaration and Power of Attorney of Inventors
- Assignment and Recordation Cover Sheet

Please see attached PTO/SB/17 for payment of fees.

Respectfully submitted,
PILLSBURY WINTHROP SHAW PITTMAN LLP

By _____
Christine H. McCarthy
Registration No. 41844
Telephone: (703) 770.7743

Date: September 26, 2006
P.O. Box 10500
McLean, VA 22102
Customer No.: 00909

Trend Micro, Inc.
EX1007-813

OIPE IAP56 SEP 26 2006 PATENT & TRADEMARK OFFICE

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371 (c) DATE	FIRST NAMED APPLICANT	ATTORNEY DOCKET NUMBER
11/477,807	06/30/2006	Melvyn Morris	011765-0355986

CONFIRMATION NO. 4785

00909
PILLSBURY WINTHROP SHAW PITTMAN, LLP
P.O. BOX 10500
MCLEAN, VA 22102

FORMALITIES LETTER

Date Mailed: 07/27/2006

NOTICE TO FILE MISSING PARTS OF NONPROVISIONAL APPLICATION

FILED UNDER 37 CFR 1.53(b)

Filing Date Granted

Items Required To Avoid Abandonment:

An application number and filing date have been accorded to this application. The item(s) indicated below, however, are missing. Applicant is given **TWO MONTHS** from the date of this Notice within which to file all required items and pay any fees required below to avoid abandonment. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a).

- The statutory basic filing fee is missing.
  Applicant must submit $ **300** *to complete the basic filing fee for a non-small entity. If appropriate, applicant may make a written assertion of entitlement to small entity status and pay the small entity filing fee (37 CFR 1.27).*
- The oath or declaration is missing. *A properly signed oath or declaration in compliance with 37 CFR 1.63, identifying the application by the above Application Number and Filing Date, is required.*
  Note: If a petition under 37 CFR 1.47 is being filed, an oath or declaration in compliance with 37 CFR 1.63 signed by all available joint inventors, or if no inventor is available by a party with sufficient proprietary interest, is required.

The applicant needs to satisfy supplemental fees problems indicated below.

The required item(s) identified below must be timely submitted to avoid abandonment:

- Additional claim fees of **$2350** as a non-small entity, including any required multiple dependent claim fee, are required. Applicant must submit the additional claim fees or cancel the additional claims for which fees are due.
- To avoid abandonment, a surcharge (for late submission of filing fee, search fee, examination fee or oath or declaration) as set forth in 37 CFR 1.16(f) of $130 for a non-small entity, must be submitted with the missing items identified in this letter.

09/27/2006 CCHAU1 00000111 033975 11477807

01 FC:1011 300.00 DA
02 FC:1111 500.00 DA
03 FC:1311 200.00 DA
04 FC:1051 130.00 DA
05 FC:1202 1350.00 DA
06 FC:1201 1000.00 DA

SUMMARY OF FEES DUE:

Total additional fee(s) required for this application is **$3480** for a Large Entity

- **$300** Statutory basic filing fee.
- **$130** Surcharge.

- The application search fee has not been paid. Applicant must submit **$500** to complete the search fee.
- The application examination fee has not been paid. Applicant must submit **$200** to complete the examination fee for a large entity

- Total additional claim fee(s) for this application is **$2350**

  - **$1000** for **5** independent claims over 3.
  - **$1350** for **27** total claims over 20.

Replies should be mailed to: Mail Stop Missing Parts
Commissioner for Patents
P.O. Box 1450
Alexandria VA 22313-1450

A copy of this notice MUST be returned with the reply.

Office of Initial Patent Examination (571) 272-4000, or 1-800-PTO-9199, or 1-800-972-6382
PART 1 - ATTORNEY/APPLICANT COPY

OIPE IAP56 SEP 26 2006 PATENT & TRADEMARK OFFICE

PTO/SB/17 (01-06)
Approved for use through 07/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

Fees pursuant to the Consolidated Appropriations Act, 2005 (H.R. 4818).

FEE TRANSMITTAL For FY 2006

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT ($) 3,520.00

Complete if Known	
Application Number	11/477,807
Filing Date	June 30, 2006
First Named Inventor	MELVYN MORRIS
Examiner Name	
Art Unit	2131
Attorney Docket No.	011765-0355986

METHOD OF PAYMENT (check all that apply)

☐ Check ☐ Credit Card ☐ Money Order ☐ None ☐ Other (please identify): _____

☒ Deposit Account Deposit Account Number: 033975 Deposit Account Name: PILLSBURY WINTHROP SHAW PITTMAN LLP

For the above-identified deposit account, the Director is hereby authorized to: (check all that apply)

☒ Charge fee(s) indicated below ☐ Charge fee(s) indicated below, **except for the filing fee**

☒ Charge any additional fee(s) or underpayments of fee(s) under 37 CFR 1.16 and 1.17 ☒ Credit any overpayments

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

FEE CALCULATION (All the fees below are due upon filing or may be subject to a surcharge.)

1. BASIC FILING, SEARCH, AND EXAMINATION FEES

	FILING FEES		SEARCH FEES		EXAMINATION FEES		
		Small Entity		Small Entity		Small Entity	
Application Type	Fee ($)	Fee ($)	Fee ($)	Fee ($)	Fee ($)	Fee ($)	Fees Paid ($)
Utility	300	150	500	250	200	100	1,000.00
Design	200	100	100	50	130	65	
Plant	200	100	300	150	160	80	
Reissue	300	150	500	250	600	300	
Provisional	200	100	0	0	0	0	

2. EXCESS CLAIM FEES

Fee Description	Fee ($)	Small Entity Fee ($)
Each claim over 20 (including Reissues)	50	25
Each independent claim over 3 (including Reissues)	200	100
Multiple dependent claims	360	180

Total Claims	Extra Claims	Fee ($)	Fee Paid ($)
47 - 20 or HP =	27	x 50.00	= 1,350.00

HP = highest number of total claims paid for, if greater than 20.

Indep. Claims	Extra Claims	Fee ($)	Fee Paid ($)
8 - 3 or HP =	5	x 200.00	= 1,000.00

HP = highest number of independent claims paid for, if greater than 3.

Multiple Dependent Claims	
Fee ($)	Fee Paid ($)

3. APPLICATION SIZE FEE

If the specification and drawings exceed 100 sheets of paper (excluding electronically filed sequence or computer listings under 37 CFR 1.52(e)), the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).

Total Sheets	Extra Sheets	Number of each additional 50 or fraction thereof	Fee ($)	Fee Paid ($)
61 - 100 =	-39	/ 50 = _____ (round up to a whole number) x	250.00 =	0.00

4. OTHER FEE(S)

	Fees Paid ($)
NLate Declaration Surchage	130.00
Other (e.g., late filing surcharge): Assignment Recordation Fee	40.00

SUBMITTED BY

Signature		Registration No. (Attorney/Agent) 41844	Telephone 703.770.7743
Name (Print/Type)	Christine McCarthy		Date September 26, 2006

This collection of information is required by 37 CFR 1.136. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 30 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

SEP 26 2006
PATENT & TRADEMARK OFFICE

FOR UTILITY
ORIGINAL
DECLARATION

RULE 63 (37 C.F.R. 1.63)
DECLARATION AND POWER OF ATTORNEY
FOR PATENT APPLICATION
IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

As a below named inventor, I hereby declare that my residence, post office address and citizenship are as stated below next to my name, and I believe I am an original, first and joint inventor of the subject matter which is claimed and for which a patent is sought on the INVENTION ENTITLED Methods and Apparatus for Dealing with Malware, the specification of which was filed on 30th June 2006 as U.S. Application No. 11/477807.

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above. I acknowledge the duty to disclose all information known to me to be material to patentability as defined in 37 C.F.R. 1.56. Except as noted below, I hereby claim foreign priority benefits under 35 U.S.C. 119(a)-(d) or 365(b) of any foreign application(s) for patent or inventor's certificate, or 365(a) of any PCT International Application which designated at least one other country than the United States, listed below and have also identified below any foreign application for patent or inventor's certificate, or PCT International Application, filed by me or my assignee disclosing the subject matter claimed in this application and having a filing date (1) before that of the application on which priority is claimed, or (2) if no priority claimed, before the filing date of this application:

PRIOR FOREIGN APPLICATION(S)

Number	Country	Filed	Date First Laid Open Or Published	Date Patented or Granted	Priority Claimed
0513375.6	GB	30th June 2005			Yes

Except as noted below, I hereby claim domestic priority benefit under 35 U.S.C. 119(e) or 120 and/or 365(c) of the indicated United States applications listed below and PCT International applications listed above or below and, if this is a continuation-in-part (CIP) application, insofar as the subject matter disclosed and claimed in this application is in addition to that disclosed in such prior applications, I acknowledge the duty to disclose all information known to me to be material to patentability as defined in 37 C.F.R. 1.56 which became available between the filing date of each such prior application and the national or PCT international filing date of this application.

PRIOR U.S. PROVISIONAL, NONPROVISIONAL AND/OR PCT APPLICATION(S)

Application Number	Filed	Status pending, abandoned, patented	Priority Claimed

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

And I hereby appoint Pillsbury Winthrop Shaw Pittman LLP, Intellectual Property Group, (to whom all communications are to be directed), and persons of that firm who are associated with USPTO Customer No. 00909 individually and collectively my attorneys to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith and with the resulting patent, and I hereby authorize them to delete from that Customer No. names of persons no longer with their firm, to add new persons of their Firm to that Customer No., and to act and rely on instructions from and communicate directly with the person/assignee/attorney/firm/organization who/which first sends/sent this case to them and by whom/which I hereby declare that I have consented after full disclosure to be represented unless/until I instruct the above Firm and/or an attorney of that Firm in writing to the contrary.

Power of Attorney to Customer Number 00909

INVENTOR'S SIGNATURE: Date:

Name	Melvyn		MORRIS
	First	Middle Initial	Family Name
Residence	Turnditch	United Kingdom	GB
	City	State/Foreign Country	Country of Citizenship
Mailing Address	Redmire Gap, Intake Lane, Turnditch, Derbyshire DE56 2LU, United Kingdom		

INVENTOR'S SIGNATURE: Date:

Name	Paul		Stubbs
	First	Middle Initial	Family Name
Residence	Wyboston	United Kingdom	GB
	City	State/Foreign Country	Country of Citizenship
Mailing Address	7 Rookery Road, Wyboston, Bedfordshire MK44 3AX, United Kingdom		

BEST AVAILABLE COPY

Trend Micro, Inc.
EX1007-817

INVENTOR'S SIGNATURE: M. Hartwig Date:

Name	Markus		HARTWIG
	First	Middle Initial	Family Name
Residence	Milton Keynes	United Kingdom	DE
	City	State/Foreign Country	Country of Citizenship
Mailing Address	Exchange House, 494 Midsummer Boulevard, Milton Keynes, Buckinghamshire MK9 2EA, United Kingdom		

INVENTOR'S SIGNATURE: D Harter Date:

Name	Darren		Harter
	First	Middle Initial	Family Name
Residence	Hucclecote	United Kingdom	GB
	City	State/Foreign Country	Country of Citizenship
Mailing Address	178 Barnwood Road, Hucclecote, Gloucester Gloucestershire GL4 3JZ, United Kingdom		

BEST AVAILABLE COPY

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371 (c) DATE	FIRST NAMED APPLICANT	ATTORNEY DOCKET NUMBER
11/477,807	06/30/2006	Melvyn Morris	011765-0355986

CONFIRMATION NO. 4785

00909
PILLSBURY WINTHROP SHAW PITTMAN, LLP
P.O. BOX 10500
MCLEAN, VA 22102

FORMALITIES LETTER

Date Mailed: 07/27/2006

NOTICE TO FILE MISSING PARTS OF NONPROVISIONAL APPLICATION

FILED UNDER 37 CFR 1.53(b)

Filing Date Granted

Items Required To Avoid Abandonment:

An application number and filing date have been accorded to this application. The item(s) indicated below, however, are missing. Applicant is given **TWO MONTHS** from the date of this Notice within which to file all required items and pay any fees required below to avoid abandonment. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a).

- The statutory basic filing fee is missing.
  Applicant must submit $ **300** *to complete the basic filing fee for a non-small entity. If appropriate, applicant may make a written assertion of entitlement to small entity status and pay the small entity filing fee (37 CFR 1.27).*
- The oath or declaration is missing. *A properly signed oath or declaration in compliance with 37 CFR 1.63, identifying the application by the above Application Number and Filing Date, is required.*
  Note: If a petition under 37 CFR 1.47 is being filed, an oath or declaration in compliance with 37 CFR 1.63 signed by all available joint inventors, or if no inventor is available by a party with sufficient proprietary interest, is required.

The applicant needs to satisfy supplemental fees problems indicated below.

The required item(s) identified below must be timely submitted to avoid abandonment:

- Additional claim fees of **$2350** as a non-small entity, including any required multiple dependent claim fee, are required. Applicant must submit the additional claim fees or cancel the additional claims for which fees are due.
- To avoid abandonment, a surcharge (for late submission of filing fee, search fee, examination fee or oath or declaration) as set forth in 37 CFR 1.16(f) of $130 for a non-small entity, must be submitted with the missing items identified in this letter.

SUMMARY OF FEES DUE:

Total additional fee(s) required for this application is **$3480** for a Large Entity

- **$300** Statutory basic filing fee.
- **$130** Surcharge.

- The application search fee has not been paid. Applicant must submit **$500** to complete the search fee.
- The application examination fee has not been paid. Applicant must submit **$200** to complete the examination fee for a large entity

- Total additional claim fee(s) for this application is **$2350**

  - **$1000** for **5** independent claims over 3.
  - **$1350** for **27** total claims over 20.

Replies should be mailed to: Mail Stop Missing Parts
Commissioner for Patents
P.O. Box 1450
Alexandria VA 22313-1450

A copy of this notice MUST be returned with the reply.

Office of Initial Patent Examination (571) 272-4000, or 1-800-PTO-9199, or 1-800-972-6382
PART 3 - OFFICE COPY

063006 17712 U.S.PTO

PTO/SB/05 (09-04)
Approved for use through 07/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office. U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket No.	011765-0355986
First Inventor	MELVYN MORRIS
Title	See attached addendum
Express Mail Label No.	

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

1. ☐ **Fee Transmittal Form** (e.g., PTO/SB/17)
   (Submit an original and a duplicate for fee processing)
2. ☐ **Applicant claims small entity status.** See 37 CFR 1.27.
3. ☒ **Specification** [*Total Pages* 58]
   Both the claims and abstract must start on a new page
   (For information on the preferred arrangement, see MPEP 608.01 (a))
4. ☒ Drawing(s) (*35 U.S.C. 113*) [*Total Sheets* 3]
5. **Oath or Declaration** [*Total Sheets*]
   a. ☐ Newly executed (original or copy)
   b. ☐ Copy from a prior application (37 CFR 1.63(d))
   (for continuation/divisional with Box 18 completed)
   i. ☐ DELETION OF INVENTOR(S)
   Signed statement attached deleting inventor(s) name in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b).
6. ☒ **Application Data Sheet.** See 37 CFR 1.76
7. ☐ **CD-ROM or CD-R** in duplicate, large table or Computer Program *(Appendix)*
   ☐ Landscape table on CD
8. **Nucleotide and/or Amino Acid Sequence Submission** *(if applicable, all necessary)*
   a. ☐ Computer Readable Form (CRF)
   b. Specification Sequence Listing on:
      i. ☐ CD-ROM or CD-R (2 copies); or
      ii. ☐ Paper
   c. ☐ Statements verifying identity of above copies

ADDRESS TO: **Commissioner for Patents**
P.O. Box 1450
Alexandria VA 22313-1450

063006 11477807 U.S.PTO 11/3261

ACCOMPANYING APPLICATION PARTS

9. ☐ **Assignment Papers** (cover sheet & document(s))
   Name of Assignee
10. ☐ **37 CFR 3.73(b) Statement** *(when there is an assignee)* ☐ Power of Attorney
11. ☐ **English Translation Document** *(if applicable)*
12. ☐ **Information Disclosure Statement** (PTO/SB/08 or PTO-1449
    ☐ Copies of Citations attached
13. ☐ **Preliminary Amendment**
14. ☐ **Return Receipt Postcard** (MPEP 503)
    (Should be specifically itemized)
15. ☐ **Certified Copy of Priority Document(s)**
    (if foreign priority is claimed)
16. ☐ **Nonpublication Request** under 35 U.S.C. 122 b)(2)(B)(i). Applicant must attach form PTO/SB/35 or equivalent.
17. ☐ Other: .....

18. If a CONTINUING APPLICATION, *check appropriate box, and supply the requisite information below and in the first sentence of the specification following the title, or in an Application Data Sheet under 37 CFR 1.76:*

☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No.: .....

Prior application information: *Examiner* _____ *Art Unit:* _____

19. CORRESPONDENCE ADDRESS

☒ The address associated with Customer Number: 00909 ***OR*** ☐ Correspondence address below

Name					
Address					
City		*State*		*Zip Code*	
Country		*Telephone*		*Fax*	

Signature		*Date*	June 30, 2006
Name (Print/Type)	Christine H. McCarthy	*Registration No. (Attorney/Agent)*	41844

This collection of information is required by 37 CFR 1.53(b). The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Addendum

Invention Title

METHODS AND APPARATUS FOR DEALING WITH MALWARE

METHODS AND APPARATUS FOR DEALING WITH MALWARE

The present invention relates generally to methods and apparatus for dealing with malware. In one aspect, the present invention relates to a method and apparatus for classifying a computer object as malware. In another aspect, the present invention relates to a method and apparatus for determining the protection that a remote computer has from malware. In another aspect, the present invention relates to a method and apparatus for classifying a computer object as malware or as safe. In another aspect, the present invention relates to a method of installing software on a computer.

The term "malware" is used herein to refer generally to any executable computer file or, more generally "object", that is or contains malicious code, and thus includes viruses, Trojans, worms, spyware, adware, etc. and the like.

A typical anti- malware product, such as anti-virus scanning software, scans objects or the results of an algorithm applied to the object or part thereof to look for signatures in the object that are known to be indicative of the presence of a virus. Generally, the method of dealing with malware is that when new types of malware are released, for example via the Internet, these are eventually detected. Once new items of malware have been detected, then the service providers in the field generate signatures that attempt to deal with these and these signatures are then released as updates to their anti-

malware programs. Heuristic methods have also been employed.

These systems work well for protecting against known malicious objects. However, since they rely on signature files being generated and/or updated, there is inevitably a delay between a new piece of malware coming into existence or being released and the signatures for combating that malware being generated or updated and supplied to users. Thus, users are at risk from new malware for a certain period of time which might be up to a week or even more. Moreover, in order to try to defeat anti-virus products, the malware writers use obfuscation techniques in order to attempt to hide the signature or signature base data of the virus code from detection. Typically, the obfuscation involves encrypting or packing the viral code.

WO-A-2004/097602 describes a system that analyses computer files received or generated by a local computer and compares these with a database of known files to determine whether a particular file is known and if so whether it has been known about long enough that it can be regarded as "safe". However, in practice, on its own this is not likely to provide for adequate protection because, for example, the active payload of a virus or Trojan may only be programmed to activate at a particular date, or upon receiving a message or instruction from a local or remote system or process, or on the occurrence of a particular event that may be many months or even years after the process has been first run or is released. Thus, just looking at the age of a file is an unsatisfactory way of determining whether it is properly safe and will remain so.

In the system of WO-A-2004/0083408, a worm in a file is detected by examining connection attempts made by the specific file running on a computer.

US-A-6944772, US-A-6772346, EP-A-1549012 and EP-A-1280040 all disclose "community-based" anti-malware systems in which a plurality of "local" computers all connect via a network (which may be a LAN or the Internet, for example) to a central computer. On encountering a file that is not already known to them, the local computers send a request to the central computer for authorisation to run the file. If the file is recognised at the central computer, then the central computer can send permission for the local computer to run the file if the file is known to be safe or send a "deny" command if the file is known to be malicious. However, in each of these prior art proposals, if the file is not known at the central computer, then the whole file is sent to the central computer where it can be analysed to determine whether it should be regarded as safe or malware. Such analysis is typically carried out manually or "semi-manually" by subjecting the file to detailed analysis, for example by emulation or interpretation, which can still take days given the human involvement that is typically required. There is therefore still a considerable period of time before a new file is classified as safe or as malware. In the case of these prior art systems, the request for authorisation to run the file that is sent by a local computer to the central computer may comprise sending a checksum or "signature" or "key" that uniquely represents the file.

A similar community-based anti-malware system is disclosed in WO-A-02/33525. In this system, in the case that a local computer is seeking clearance to run a file that is not known by the central computer to be safe or malware, some limited audit information about the prevalence of the file on other local computers can be sent to a human system administrator associated with the local computer that is seeking permission to run the file. The human system administrator can therefore make a better informed though still "manual" decision as to whether or not the file is safe to run.

In the system of US-A-2004/0073810, a metafile containing data about an attachment or other transmitted file is sent to a central computer. The data about that file is analysed to determine a likelihood of the transmitted file being malware. A specific example given is that if the transmitted file has been transmitted via at least a certain number of servers, then it should be treated as malware.

In the systems disclosed in US-A-2005/0021994 and US-A-2004/0153644, pre-approved files, which may be certified as safe by for example the software vendor associated with the files, may be permitted always to run without further checking. In one embodiment of the system of US-A-2004/0153644, monitoring is carried out to decide that a file is malicious if an abnormally high number of requests by that file is received at a central authority from plural local computers in a time period or if an abnormally high number of requests by that file on a single computer is received from the single local computer in a time period.

In the system of US-A-2004/0006704, a comparison is made between installed versions of software on a computer with a database of software versions and their known vulnerabilities. A user of the computer can therefore be informed of specific risks and how to minimise those risks by updating existing or installing new software.

In the system of WO-A-03/021402, a central database holds a virtual image of all files stored on each of plural local computers. If a threat in one local computer is identified, other local computers with a similar configuration can be notified of the risk.

Thus, the prior art systems either rely on deep analysis of a new object in order to determine whether or not the object is malicious, which introduces delay and therefore risk to users during the period that the file is analysed and new anti-malware signatures distributed, or limited analysis of the operation of the particular object or its method of transmission to a computer is carried out to decide a likelihood of the object being malicious.

According to a first aspect of the present invention, there is provided a method of classifying a computer object as malware, the method comprising:

at a base computer, receiving data about a computer object from each of plural remote computers on which the object or similar objects are stored;

comparing in the base computer the data about the computer object received from the plural computers; and,

classifying the computer object as malware on the basis of said comparison.

Compared to the prior art that relies solely on signature matching, this aspect allows a comparison to be made between the objects and/or their effects on the different remote computers to determine whether or not a particular object should be classed as good or as malware. Sophisticated pattern analysis can be carried out. This allows a rapid determination of the nature of the object to be made, without requiring detailed analysis of the object itself as such to determine whether it malware and also avoids the need to generate new signatures to be used for signature matching as in the conventional prior art anti-virus software.

In a preferred embodiment, the data about the computer object that is sent from the plural remote computers to the base computer and that is used in the comparison includes one or more of: executable instructions contained within or constituted by the object; the size of the object; the current name of the object; the physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers.

Preferably, the comparing identifies relationships between the object and other objects. In an example, this can be used immediately and automatically to mark a child object as bad (or good) if the or a parent or other related object is bad (or good). Thus, if at least one other

object to which said object is related is classed as malware, then the method may comprise classifying said object as malware. Said other objects include the object or similar objects stored on at least some of the remote computers. Said other objects may include other objects that are parent objects or child objects or otherwise process-related objects to said object.

In a most preferred embodiment, the data is sent in the form of key that is obtained by a hashing process carried out in respect of the objects on the respective remote computers. A major advantage of using such a key is that it keeps down the volume of data that needs to be transmitted to the base computer. Given that there may be thousands or even millions of connected remote computers and further given that each may send details about very many objects, this can be an important advantage.

The key preferably has at least one component that represents executable instructions contained within or constituted by the object. This important preferred feature allows a comparison to be made at the base computer of only the executable instructions of the object. This means for example that differently named objects that basically have the same executable instructions, which is often an indicator that the objects are malware, can nevertheless be regarded as the "same" object for this purpose. As another example, a new version of a program may be released which has minor changes compared to a previous version already known to the base computer and which in substance, at least in respect of the executable instructions, can be regarded as being the same as the previous version. In that case, the minor differences can

be ignored and the objects regarded as being the same. Not only is this useful in distinguishing between malware and for example revised versions of previous software, it also keeps down the data transmission and storage requirements because the base computer can inform the remote computers that an apparently new object is for this purpose the same as a previously known object, thus avoiding having the remote computers send full details about the object or the object itself to the base computer.

The key preferably has at least one component that represents data about said object. Said data about said object may include at least one of: the current name of the object; the physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers.

The key preferably has at least one component that represents the physical size of the object.

Where more than one of these components are present in the key, the plural components are preferably severable.

The method may comprise initially classifying an object as not malware, generating a mask for said object that defines acceptable behaviour for the object, and comprising monitoring operation of the object on at least one of the remote computers and reclassifying the object as

malware if the actual monitored behaviour extends beyond that permitted by the mask. This provides an efficient and effective way of monitoring the behaviour of an object that has been classified or regarded as good and allows the object to be reclassified quickly as malware if the behaviour of the object warrants it.

According to a second aspect of the present invention, there is provided apparatus for classifying a computer object as malware, the apparatus comprising:

a base computer constructed and arranged to receive data about a computer object from each of plural remote computers on which the object or similar objects are stored;

the base computer being constructed and arranged to compare the data about the computer object received from said plural computers; and,

the base computer being constructed and arranged to classify the computer object as malware on the basis of said comparison.

According to a third aspect of the present invention, there is provided a method of providing data about a computer object from a remote computer to a base computer so that a comparison can be made at the base computer with similar data received from other remote computers, the method comprising:

providing from a remote computer to a base computer data about a computer object that is stored on the remote computer;

the data including one or more of: executable instructions contained within or constituted by the object; the size of the object; the current name of the object; the

physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers;

the data being sent in the form of key that is obtained by a hashing process carried out in respect of the object on the remote computer.

This method, which may be carried out by so-called agent software running on the remote computer, allows for efficient sending of data to the base computer, which minimises data transmission and storage requirements and also permits rapid analysis to be made at the base computer.

The key preferably has at least one component that represents executable instructions contained within or constituted by the object.

The key preferably has at least one component that represents data about said object. Said data about said object may include at least one of: the current name of the object; the physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the

object is created, configured or runs on the respective remote computers.

The key preferably has at least one component that represents the physical size of the object.

According to a fourth aspect of the present invention, there is provided a method of determining the protection that a remote computer has from malware, the method comprising:

receiving at a base computer details of all or selected security products operating at a point in time on said remote computer;

receiving similar information from other remote computers connected to the base computer; and,

identifying any malware processes that were not identified by said other remote computers having that particular combination of security products.

In this way, the base computer can be used to obtain information as to whether for example a particular, specific combination of operating system and various security products, including settings and signature files existing at a point in time, renders a particular computer having those products and settings susceptible or vulnerable to any particular malware object. The user can be advised accordingly and for example provided with recommendations for remedying the situation.

The method may therefore comprise providing information to the user of said remote computer that said remote computer may be susceptible to attack by said malware processes on the basis of said identifying.

The details of all or selected security products preferably includes the name of the security products, versions, and loaded signature files.

According to a fifth aspect of the present invention, there is provided apparatus for determining the protection that a remote computer has from malware, the apparatus comprising:

a base computer constructed and arranged to receive computer details of all or selected security products operating at a point in time on said remote computer;

the base computer being constructed and arranged to receive similar information from other remote computers connected to the base computer; and,

the base computer being constructed and arranged to identify any malware processes that were not identified by said other remote computers having that particular combination of security products.

According to a sixth aspect of the present invention, there is provided a method of classifying a computer object as malware or as safe, wherein said computer object is a descendant or otherwise related object of a first computer object, the method comprising:

classifying a first computer object as malware or as safe;

identifying in a key relating to said first computer object a component that uniquely identifies the first computer object and that is inherited or otherwise present in the key of a descendant or other related computer object of the first computer object; and,

classifying said computer object as malware or as safe as the case may be on the basis of the unique identifier component being present in the key of said computer object.

This aspect uses the concept of ancestry to enable objects to be marked as malware. For example, any particular process may spawn child processes which are therefore related. The key relating to the first object may be inspected to identify a component that uniquely identifies the first object and that is inherited or otherwise present in the key of a descendant or other related object of the first object.

The method may comprise monitoring activities of said first computer object and reclassifying the first computer object as malware in the case that it was initially classified as safe and subsequently determined to be malware, the method further comprising automatically classifying as malware any computer object that has a key in which said unique identifier component is present.

According to a seventh aspect of the present invention, there is provided apparatus for classifying a computer object as malware or as safe, wherein said computer object is a descendant or otherwise related object of a first computer object, the apparatus comprising:

a computer constructed and arranged to classify a first computer object as malware or as safe;

the computer being constructed and arranged to identify in a key relating to said first computer object a component that uniquely identifies the first computer object and that is inherited or otherwise present in the

key of a descendant or other related computer object of the first computer object; and,

the computer being constructed and arranged to classify said computer object as malware or as safe as the case may be on the basis of the unique identifier component being present in the key of said computer object.

According to an eighth aspect of the present invention, there is provided a method of installing software on a computer, the method comprising:

on initiation of installation of software on a computer, providing a computer-generated prompt on the computer to a user to ascertain whether the user authorises the installation; and,

ceasing the installation if a user authorisation is not received, else:

receiving at the computer the user's authorisation to proceed with the installation;

proceeding with the installation;

obtaining data about computer objects that are created or used during the installation;

storing said data at the local computer.

This provides for security when a user is installing new software and is not for example connected to a base computer having a community database of the type mentioned above. In that case, the method, which may be implemented in agent software running on the local computer, allows the user to permit the installation to proceed whilst at the same time gathering data about the objects (such as processes, new files, etc.) that are created during the installation.

Preferably, the locally stored data is referred to during the installation to ensure that all objects created or used during the installation are part of the installation process, and, if it is found that objects created or used during the installation are not part of the installation process, either or both of: (i) ceasing the installation and (ii) providing a computer-generated prompt on the computer to the user accordingly. This allows the method to ensure that only those objects that are required for the installation are permitted to be created or used and thus avoids unwittingly allowing malware to install (given that malware often creates objects that are not expected as part of a normal installation of new software).

In a preferred embodiment, the method comprises connecting the computer to a community database that is connectable to a plurality of computers, and uploading the stored data to the community database for comparison with similar data provided by other computers.

The method may comprise downloading data about trusted installers to the computer, said data about trusted installers being referred to during the installation such that any objects relating to or created by the trusted installer are automatically authorised to proceed. This facilitates installation of software that is known a priori to be trustworthy.

Said data about trusted installers may be referred to only for a predetermined time period following receipt at the computer of the user's authorisation to proceed with the installation.

The present invention also includes computer programs comprising program instructions for causing a computer to perform any of the methods described above.

Although the embodiments of the invention described with reference to the drawings comprise computer processes performed in computer apparatus and computer apparatus itself, the invention also extends to computer programs, particularly computer programs on or in a carrier, adapted for putting the invention into practice. The program may be in the form of source code, object code, a code intermediate source and object code such as in partially compiled form, or in any other form suitable for use in the implementation of the processes according to the invention. The carrier be any entity or device capable of carrying the program. For example, the carrier may comprise a storage medium, such as a ROM, for example a CD ROM or a semiconductor ROM, or a magnetic recording medium, for example a floppy disk or hard disk. Further, the carrier may be a transmissible carrier such as an electrical or optical signal which may be conveyed via electrical or optical cable or by radio or other means.

Embodiments of the present invention will now be described by way of example with reference to the accompanying drawings, in which:

Figure 1 shows schematically apparatus in which an embodiment of the present invention may be implemented;

Figure 2 is a flowchart showing schematically the operation of an example of a method according to an embodiment of the present invention;

Figure 3 is a flowchart showing schematically the operation of another example of a method according to an embodiment of the present invention; and,

Figure 4 is a flowchart showing schematically an information obtaining stage.

Referring to Figure 1, a computer network is generally shown as being based around a distributed network such as the Internet 1. The present invention may however be implemented across or use other types of network, such as a LAN. Plural local or "remote" computers 2 are connected via the Internet 1 to a "central" or "base" computer 3. The computers 2 may each be variously a personal computer, a server of any type, a PDA, mobile phone, an interactive television, or any other device capable of loading and operating computer objects. An object in this sense may be a computer file, part of a file or a sub-program, macro, web page or any other piece of code to be operated by or on the computer, or any other event whether executed, emulated, simulated or interpreted. An object 4 is shown schematically in the figure and may for example be downloaded to a remote computer 2 via the Internet 1 as shown by lines 5 or applied directly as shown by line 6.

In one preferred embodiment, the base computer 3 holds a database 7 with which the remote computers 2 can interact when the remote computers 2 run an object 4 to determine whether the object 4 is safe or unsafe. The community database 7 is populated, over time, with information relating to each object run on all of the connected remote computers 2. As will be discussed further below, data

representative of each object 4 preferably takes the form of a so-called signature or key relating to the object and its effects. As will also be discussed further below, the database 7 may further include a mask for the object 4 that sets out the parameters of the object's performance and operation.

Referring now to Figure 2, at the start point 21, a computer object 4 such as a process is run at a remote computer 2. At step 22, by operation of local "agent" software running on the remote computer 2, the operation of the process is hooked so that the agent software can search a local database stored at the remote computer 2 to search for a signature or key representing that particular process, its related objects and/or the event. If the local signature is present, it will indicate either that the process is considered to be safe or will indicate that that process is considered unsafe. An unsafe process might be one that has been found to be malware or to have unforeseen or known unsafe or malevolent results arising from its running. If the signature indicates that the process is safe, then that process or event is allowed by the local agent software on the remote computer 2 to run at step 23. If the signature indicates that the process is not safe, then the process or event is stopped at step 24.

It will be understood that there may be more than two states than "safe" or "not-safe" and choices may be given to the user. For example, if an object is considered locally to be not safe, the user may be presented with an option to allow the related process to run nevertheless. It is also possible for different states to be presented to each remote computer 2. The state can be varied by the

central system to take account of the location, status or ownership of the remote computer or timeframe.

If the object is unknown locally, then details of the object are passed over the Internet 1 or other network to the base computer 3 for storing in the community database 7 and preferably for further analysis at the base computer 3. In that case, the community database 7 is then searched at step 25 for a signature for that object that has already been stored in the community database 7. The community database 7 is supplied with signatures representative of objects, such as programs or processes, run by each monitored remote computer 2. In a typical implementation in the field, there may be several thousands or even millions of remote computers 2 connected or connectable to the base computer 3 and so any objects that are newly released upon the Internet 1 or that otherwise are found on any of these remote computers 2 will soon be found and signatures created and sent to the base computer 3 by the respective remote computers 2.

When the community database 7 is searched for the signature of the object that was not previously known at the remote computer 2 concerned, then if the signature is found and indicates that that object is safe, then a copy of the signature or at least a message that the object is safe is sent to the local database of the remote computer 2 concerned at step 26 to populate the local database. In this way, the remote computer 2 has this information immediately to hand the next time the object 4 is encountered. A separate message is also passed back to the remote computer 2 to allow the object to run in the current instance.

If the signature is found in the community database 7 and this indicates for some reason that the object is unsafe, then again the signature is copied back to the local database and marked "unsafe" at step 27, and/or a message is sent to the remote computer 2 so that running of the object is stopped (or it is not allowed to run) and/or the user given an informed choice whether to run it or not.

If after the entire community database 7 has been searched the object is still unknown, then it is assumed that this is an entirely new object which has never been seen before in the field. A signature is therefore created representative of the object at step 28, or a signature sent by the remote computer 2 is used for this purpose, and this signature is initially marked as bad or unsafe community database 7 at step 29. The signature is copied to the local database of the remote computer 2 that first ran the object at step 30. A message may then be passed to the remote computer 2 to instruct the remote computer 2 not to run the object or alternatively the user may be given informed consent as to whether to allow the object to run or not. In addition, a copy of the object itself may be requested at step 31 by the community database 7 from the remote computer 2.

If the user at the remote computer 2 chooses to run a process that is considered unsafe because it is too new, then that process may be monitored by the remote computer 2 and/or community database 7 and, if no ill effect occurs or is exhibited after a period of time of n days for example, it may then be considered to be safe. Alternatively, the community database 7 may keep a log of each instance of the

process which is found by the many remote computers 2 forming part of the network and after a particular number of instances have been recorded, possibly with another particular number of instances or the process being allowed to run and running safely, the signature in the community database 7 may then be marked as safe rather than unsafe. Many other variations of monitoring safety may be done within this concept.

The details of an object 4 that are passed to the base computer 3 are preferably in the form of a signature or "key" that uniquely identifies the object 4. This is mainly to keep down the data storage and transmission requirements. This key may be formed by a hashing function operating on the object at the remote computer 2.

The key in the preferred embodiment is specially arranged to have at least three severable components, a first of said components representing executable instructions contained within or constituted by the object, a second of said components representing data about said object, and a third of said components representing the physical size of the object. The data about the object in the second component may be any or all of the other forms of identity such as the file's name, its physical and folder location on disk, its original file name, its creation and modification dates, vendor, product and version and any other information stored within the object, its file header or header held by the remote computer 2 about it; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers. In general, the information provided in the key may include at least one of these

elements or any two or more of these elements in any combination.

In one preferred embodiment, a check sum is created for all executable files, such as (but not limited to) .exe and .dll files, which are of the type PE (Portable Executable file as defined by Microsoft). Three types of checksums are generated depending on the nature of the file:

Type 1: five different sections of the file are check summed. These include the import table, a section at the beginning and a section at the end of the code section, and a section at the beginning and a section at the end of the entire file. This type applies to the vast majority of files that are analysed;

Type 2: for old DOS or 16 bit executable files, the entire file is check summed;

Type 3: for files over a certain predefined size, the file is sampled into chunks which are then check summed. For files less than a certain predefined size, the whole file is check summed.

For the check summing process, in principle any technique is possible. The MD5 (Message-Digest algorithm 5) is a widely-used cryptographic hash function that may be used for this purpose.

This allows a core checksum to be generated by viewing only the executable elements of the checksum and making a

comparison between two executables that share common executable code.

For the type 1 checksum mentioned above, three signature processes may be used. The first defines the entire file and will change with almost any change to the file's content. The second attempts to define only the processing instructions of the process which changes much less. The third utilises the file's size, which massively reduces the potential of collisions for objects of differing sizes. By tracking the occurrences of all signatures individually appearing with different counterparts, it is possible to identify processes that have been changed or have been created from a common point but that have been edited to perform new, possibly malevolent functionality.

This "meta data" enables current and newly devised heuristics to be run on the data in the community database 7.

The data stored in the community database 7 provides an extensive corollary of an object's creation, configuration, execution, behaviour, identities and relationships to other objects that either act upon it or are acted upon by it.

The preferred central heuristics use five distinct processes to establish if an object is safe, unsafe or suspicious.

The first of the said processes utilises the singularity or plurality of names, locations, vendor,

product and version information captured and correlated from all of the remote computers 2 that have seen the object. By considering the plurality of this information for a single object, a score can be determined which can be used as a measure of the authenticity and/or credibility of the object. Most safe objects tend not to use a large plurality of identifying information or locations. Rules can be established to consider this information in respect of the type of object and its location. For example, temporary files often utilise a plurality of system generated file names which may differ on each remote computer for the same object. Where an object has little plurality, then it provides a reference point to consider its behaviour in comparison to the known behaviours of other objects that have previously used that identifying information. For example, a new object that purports to be a version of notepad.exe can have its behaviour compared with the behaviour of one or more other objects that are also known as notepad.exe. This comparison may be against a single other object or multiple other objects that use the same or even similar identifying information. In this way, new patterns of behaviour can be identified for the new object. Also it allows the preferred embodiment to police an object's behaviour over time to identify new behavioural patterns that may cause an object that was previously considered safe to have its status reconsidered. Alternatively, the score based on identities may be considered along with scores for other objects or scores created by other processes on this object to be considered in combinations.

The second of the said processes utilises an object's relationship to other objects that act upon it or upon

which it acts. For example, analysis can be made of which object created this object, which objects this object created, which objects created a registry key to run or configure this object, which objects were configured by or had registry keys created by this object, etc. In this regard an object is considered to have a relationship based on the event performed by it upon another object, or upon it by another object. This simple 1-to-1 relationship chain provides a complex series of correlation points, allowing ancestral relationships to be considered for any object and its predecessors by event or its issue (i.e. child and sub-child processes) by event. This allows a score to be developed that describes its relationships and associations with known, unknown, known safe or known bad objects or a combination thereof. Objects that have specific relationships, volumes of relationships or mixes of relationships to one type or another may be judged safe or unsafe accordingly. Alternatively, the relationship-based score may be considered along with other scores to arrive at a determination of safe or unsafe. This data can also be used to deduce a number of factors about objects related directly or via other objects and their behaviours. For example it is possible to deduce how one object's behaviour can be influenced or changed by its association or linkage to another. Consider for example notepad.exe as supplied by Microsoft with the Windows series of operating systems. It has a limited range of functionality and would not be expected therefore to perform a wide variety of events, such as transmitting data to another computer or running other programs etc. However, the behaviour of notepad.exe could be modified by injecting new code into it, such as via dynamic link library injection (DLL injection). In this case notepad.exe would now have new

capabilities derived by the code injection or linkage to another object. Using the data that defines the relationships between objects it is possible to deduce that the new behaviours of a program can be attributed to the association with another object. If that new behaviour is malevolent, then it is possible to mark either or all processes as unsafe as appropriate.

The combination of behaviours captured provide a basis to determine if the object is safe or unsafe. Malware typically exhibit certain behaviour and characteristics. For example, malware frequently has a need to self-persist. This manifests itself in the need to automatically restart on system restarts or upon certain events. Creating objects in specific locations to auto restart or trigger execution is a typical characteristic of malware. Replacing core objects of the Windows system environment are another example of typical characteristics of malware. By providing a pattern of behaviour, the determination of objects to be unsafe or safe can be automated. The centralisation of the community data in the community database 7 provides the ability to rapidly assimilate object behaviours, allowing for the rapid identification and determination of malware. Objects may also perform events upon themselves which can be considered in deriving a score.

The third said process involves time and volumes. Relevant data includes when the object was first seen, when it was last seen, how many times it has been seen, how many times it has been seen in a given interval of time, and the increase or decrease of acceleration in it being seen. This information is highly relevant in determining the

prevalence of an object in the community of remote computers. A score is developed based on these metrics which can be used to determine if an object is safe, unsafe or too prevalent to allow it to execute or propagate without very thorough examination. In this case, the object can be temporarily held or blocked from executing pending further information about its behaviour or relationships. This score may also be used in combination with scores from other processes. Time is also highly relevant in combination with other information, including but not limited to behaviours and identities. For example in the case of polymorphic or randomly named objects, time is a powerful qualifier. (A polymorphic virus changes its encryption algorithm and the corresponding encryption keys each time it replicates from one computer to another and so can be difficult to detect by conventional measures.) A program that creates other programs can often be considered normal or abnormal based on its activity over time.

The fourth said process considers the behaviour of an object. This allows a score to be developed based on the types of events performed by an object or events performed on it by itself or other objects. The centralised system of the community database 7 allows for an unlimited number of event types and can consider the object performing the event or the object having the event performed upon it, or both. Some event types also relate to external information other than objects, for example a program performing an event to connect with an Internet Chat Relay site, or a program modifying a non-executable file such as the Windows hosts file. The behavioural events of an object, be they as "actor" (i.e. the object doing something to another object) or as "victim" (i.e. the object has something done

to it by another object) of any event can be considered in many ways, such as in combination, in sequence, in volume, in presence or absence, or in any combination thereof. The behavioural events in the preferred embodiment may have been provided by a remote computer 2 or from other external sources. The process can consider these in isolation or in combination. Furthermore it is a feature of the preferred embodiment that the behavioural events can be considered in combination with the status of other objects upon which the object acts or that act upon the object. For example, creating a program may have a different score if the program being created is safe, unsafe, new, unknown or suspicious. Similarly, a program that is created by a known bad program will likely have a different score attributed to its creation event depending on the status of the object creating it.

The fifth process considers the behaviour of a web page or script. In this respect, the web page and url combination is assigned a unique identity which allows its behaviour to be tracked as if it were an object like any other executable file. In this example, the web page may perform events that would normally be seen as events performed by the web browser (e.g. IExplore.exe or Firefox.exe). The preferred embodiment substitutes the identifying details and signatures of the web browser for the "pseudo" object identity associated with the web page being displayed or executing within the browser. In this respect, the status of the web page and/or web site to which it relates may be determined as safe, unsafe, unknown or suspicious in the same way as any other object. The web page's "pseudo" object identity also allows the preferred embodiment to block, interject or limit the functionality

of that web page or web site to prevent some or all of its potentially unsafe behaviour or to provide the remote user with qualifying information to guide them about the safety of the web site, web page or their content.

Amongst other types, the types of meta data captured might be:

"Events": these define the actions or behaviours of an object acting upon another object or some other entity. The event has three principal components: the key of the object performing the act (the "Actor"), the act being performed (the "Event Type"), and the key of the object or identity of an other entity upon which the act is being performed (the "Victim"). While simple, this structure allows a limitless series of behaviours and relationships to be defined. Examples of the three components of an event might be:

Actor	Event Type	Victim
Object 1	Creates Program	Object 2
Object 1	Sends data	IP Address 3
Object 1	Deletes Program	Object 4
Object 1	Executes	Object 2
Object 2	Creates registry key	Object4

"Identities": these define the attributes of an object. They include items such as the file's name, its physical location on the disk or in memory, its logical location on the disk within the file system (its path), the file's header details which include when the file was created, when it was last accessed, when it was last modified, the information stored as the vendor, the product

it is part of and the version number of the file and it contents, its original file name, and its file size.

"Genesisactor" - the key of an object that is not the direct Actor of an event but which is the ultimate parent of the event being performed. For example in the case of a software installation, this would be the key of the object that the user or system first executed and that initiated the software installation process, e.g. Setup.exe.

"Ancillary data": many events may require ancillary data, for example an event such as that used to record the creation of a registry run key. In this situation the "event" would identify the Actor object creating the registry run key, the event type itself (e.g. "regrunkey"), and the Victim or subject of the registry run key. The ancillary data in this case would define the run key entry itself; the Hive, Key name and Value.

"Event Checksums": because the event data can be quite large extending to several hundred bytes of information for a single event, its identities for the Actor and Victim and any ancillary data, the system allows for this data itself to be summarised by the Event Checksums. Two event checksums are used utilising a variety of algorithms, such as CRC and Adler. The checksums are of the core data for an event. This allows the remote computer 2 to send the checksums of the data to the central computer 3 which may already have the data relating to those checksums stored. In this case, it does not require further information from the remote computer 2. Only if the central computer 3 has never received the checksums will it request the associated data from the

remote computer 2. This affords a considerable improvement in performance for both the remote and central computers 2,3 allowing much more effective scaling.

Thus, the meta data derived from the remote computers 2 can be used at the community database 7 to define the behaviour of a process across the community. As mentioned, the data may include at least one of the elements mentioned above (file size, location, etc.) or two or three or four or five or six or all seven (or more elements not specifically mentioned here). This may be used accordingly to model, test and create new automated rules for use in the community database 7 and as rules that may be added to those held and used in the local database of the remote computers 2 to identify and determine the response of the remote computers 2 to new or unknown processes and process activity.

Moreover, it is possible to monitor a process along with any optional sub-processes as an homogenous entity and then compare the activities of the top level process throughout the community and deduce that certain, potentially malevolent practices only occur when one or more specific sub-processes are also loaded. This allows effective monitoring (without unnecessary blocking) of programs, such as Internet Explorer or other browsers, whose functionality may be easily altered by downloadable optional code that users acquire from the Internet, which is of course the principal source of malevolent code today.

The potentially high volume of active users gives a high probability of at least one of them being infected by new malware. The speed of propagation can be detected and

recorded so that the propagation of malware can be detected and malware designated as bad on the basis of the speed of propagation, optionally in combination with the other factors discussed above, such as file size, location and name. The simple volume of infection can also be used as a trigger. In a further embodiment, difference of naming of an otherwise identical piece of code combined with acceleration of first attempts to execute the code within the community allows pattern matching that will show up an otherwise identically signatured piece of code as bad.

This feature allows the statement in some embodiments that "nothing will propagate in our community faster than X without being locked down", so that if any process or event propagates more quickly over a given duration, it is marked as bad. This is for reasons of safety given that if for example an object is propagating quickly enough, then it might infect computers before it can be analysed to determine whether or not it is malware.

This process can be automated by the identification of the vector of propagation in the community (i.e. the source of type of propagation), from timestamp data held in the community database and the marking of a piece of code that has these attributes as bad. By comparison, it is believed that all other anti-malware providers rely on a simplistic known bad model and therefore are reliant primarily on malware infection actually occurring on terminals and being reported.

Thus, the community database 7 can be used to make early diagnosis, or simply to take precautionary measures, and thus stop potentially fast propagating worms and other

malware very, very early in their life cycle. Given that it is possible to create a worm that can infect every computer connected to the Internet within a matter of a few minutes, this feature is highly desirable.

Even faster determination may be made by combining data defining the speed of propagation of a new piece of software with metadata collected by the agent software from the remote computers 2 and fed to the community database 7. This includes monitoring processes that attempt to conceal themselves from the user by randomly changing name and/or location on the remote computers 2. It also includes a process's attempt to create an identical copy (i.e. with identical code contents) on the computer but with a different name. This is a classic attribute of a worm.

The signature of an object may comprise or be associated with a mask which can be built up with use of that object and which indicates the particular types of behaviour to be expected from the object. If an object is allowed to run on a remote computer 2, even if the initial signature search 22 indicates that the object is safe, then operation of that object may be monitored within the parameters of the mask. The mask might indicate for example, the expected behaviour of the object; any external requests or Internet connections that that object might legitimately have to make or call upon the remote computer 2 to make, including details of any ports or interfaces that might be required to be opened to allow such communication; any databases, either local or over a local area network or wide area network or Internet, that may be expected to be interrogated by that object; and so on.

Thus, the mask can give an overall picture of the expected "normal" behaviour of that object.

In practice, therefore, in one embodiment the behaviour of the object is continually monitored at the remote computer(s) 2 and information relating to that object continually sent to and from the community database 7 to determine whether the object is running within its expected mask. Any behaviour that extends beyond the mask is identified and can be used to continually assess whether the object continues to be safe or not. Thus, if for example the object, on a regular basis (say monthly or yearly) opens a new port to update itself or to obtain regular data, then this information is flagged. If it is found that the object has done this on other remote computers and has had no ill effects, or this behaviour is known from other objects and known to be safe, then this behaviour might be considered as safe behaviour and the mask is then modified to allow for this. If it has been found previously that this new behaviour in fact causes unsafe or malevolent results, then the object can then be marked as unsafe even if previously it was considered safe. Similarly, if the object attempts to connect to a known unsafe website, database or to take action that is known as generally being action only taken by unsafe programs, then again the object may be considered to be unsafe.

This is shown schematically in Figure 3. Figure 3 also shows the concept that any object can be pre-authorised by, for example a trusted partner, such as a major software company, a verification authority, a Government department, and so on. Pre-authorisation enables a supplier of a new object, which has not been

released before, to get pre-authorisation for that object, and optionally includes the provision by that supplier of a mask detailing the expected and allowable behaviour of that object.

Referring to Figure 3 for example, when a process is run, the local and/or community databases are searched as before at step 31. If the process is not a pre-authorised one, then the steps of Figure 2 may be taken and the process might be allowed to run or not at step 32. If the process is pre-authorised, as determined at step 33, then it is immediately allowed to run, step 34. This may terminate the operation of the method. However, in a preferred variation, the process is then monitored whilst running, and is monitored each time it is run in the future in a monitoring state step 35 to determine whether its behaviour falls within its pre-authorised mask 36. If the behaviour falls within the pre-authorised behaviour, then the process is allowed to continue to run. If the behaviour extends beyond the allowed mask, such as by trying to instigate further processes or connections that have not been pre-authorised, then this behaviour is flagged at an alert step 37. Various actions could be taken at this stage. The process might simply not be allowed to run. Alternatively, the trusted authority that initially enabled pre-authorisation might be contacted, who may be able to confirm that this behaviour is acceptable or not. If it is acceptable, then the mask could be modified accordingly. If not acceptable, then the process might be marked as unsafe. Many other actions may be taken upon the noting of such an alert state.

If the process has been found not to be pre-authorised at step 33 but is nevertheless allowed to run, then the process is monitored at step 38 in order to generate a mask 39 representative of the normal behaviour of that process. Data representative of this mask might be sent to the community database 7 for scanning when other computers run that process. By continually monitoring a process each time it is run or during running of the process, any behaviour that differs from previous behaviour of the process can be noted and the mask can be modified, or this behaviour might be used to determine that a process that was once considered safe should now be considered to be unsafe.

In another embodiment, a computer 2 may have agent software installed that periodically or on-demand provides information to the community database 7 that is representative of all or selected ones of the software products loaded on or available the computer 2. In particular, this may be information on one or more of: all the locally-loaded security products (such as anti-malware systems including anti-virus software, anti-spyware, anti-adware and so on), firewall products, specific settings and details of which signature files are currently loaded, version details for the operating system and other software, and also information such as which files are operating and the particular version and software settings at any time. (It will be understood from the following that auditing and testing for a match for more of these criteria increases the likelihood of computers being very similarly arranged and thus reduces the rate of false negatives and positives during the match search.)

The information relating to these software products, etc. may be provided individually to the community database. Preferably however, again for reasons of data quantity for storage and transmission, the information is provided as a signature or key representative of the information (e.g. by a hashing or compression function at the computer 2). Figure 4 shows schematically how the details of all local security products, versions, signature files, firewall settings, etc. 40 are used to create a key 41. The key is transmitted to the community database 7. Since the community database 7 is provided with such information from many, possibly millions, of users' computers 2, it is likely to hold corresponding information for other computers 2 that have the same or a similar configuration of security products, etc. Thus, the community database 7 can be searched at step 42 for other computers 2 having the same or a similar combination of security products including the same setting, signature files loaded and so on.

The community database 7 in this embodiment is also provided by the agent software with details of processes run by every computer 2 and thus knows whether or not a process has been detected by each computer 2.

In this way, the community database 7 can be used to obtain information as to whether for example a particular, specific combination of operating system and various security products, including settings and signature files existing at a point in time, renders a particular computer 2 having those products and settings susceptible or vulnerable to any particular malware object.

In a simple example, if for example the database knows that a computer in the past has version A of anti-virus product B with downloaded signature update C, and also has a firewall D with particular settings E, and perhaps anti-spyware software F with signature updates G, but that a particular malware process P was not detected by this combination of programs at that point in time, then this information can be provided to a computer 2 that is known as having that combination of security programs/settings, and can be used to indicate that that computer 2 is vulnerable in the short term to attack by that particular malware process. This information can be presented to the user either by displaying a window 43 on the screen display of the computer 2 or by directing the user to a particular website which explains the position in more detail. The user might be informed for example that their particular combination of security products, etc., exposes their computer to a risk of being infected by the Sobig virus as that virus is not detectable by their computer. The user might be offered specific advice (e.g. to update a particular anti-virus program with a particular signature file) or software to download and install to remove the risk.

Thus, the community database 7, when provided with information relating to all the security products, etc. on a particular computer at a particular time, is searched for events for processes marked as "bad" that occurred on computers with that particular mix of security products and that were not locally detected. This information can then be fed back to the user of the particular computer, for example directly or by directing the user to a website. This information can be provided virtually in real-time,

allowing a new user or a user of a new computer to be able to increase the computer's effective security very quickly.

The preferred method also tracks which objects are related to each other and uses the concept of ancestry to enable objects to be marked as malware. For example, any particular process may spawn child processes which are therefore related. The key relating to the first object may be inspected to identify a component that uniquely identifies the first object and that is inherited or otherwise present in the key of a descendant or other related object of the first object. This component is referred to herein as a "gene". This general technique may be used in a number of ways:

a) A known and trusted parent process is afforded the ability to create child processes which may be automatically marked as safe to run on the local computer. It is also possible that this "inherited" property may be passed down to grand children processes and so on. This safe status is passed to the parent's child processes and possibly, through them, further child processes (referred to here as "issue"), such signatures for the issue can all automatically be recorded in the local database as good. This allows the issue processes to be quickly marked as good, even if a connection to the community database 7 is not available.

b) By monitoring activity of the parent process, if it is later found that the parent process is malware, then all of the issue processes can all automatically be recorded in the local database as bad.

C) Similarly, by monitoring activity of the issue processes, if it is later found that one of the issue processes is malware, then one or more of the parent process and all of the other issue processes (i.e. all of the related processes in this context) can all automatically be recorded in the local database as bad.

d) Parental creation of a signature for a child or children including the ability for these to be automatically marked as either good or bad depending on the parent's behaviour and determination. Note that in some embodiments the product can "watch" or monitor the birth of a child process and automatically create the signature upon arrival. This provides the ability to monitor the creation of a bad program by another bad program. It is possible therefore to monitor the ancestry of a program so if for example the grandfather creates a program (the father) and this in turn creates a bad program (the son), it is possible automatically to determine the father as a bad program.

e) A feature may be included that allows for automatic forfeiture of a child's inherited ability to trigger the automatic creation of signatures on any further births because the child, as parent, has already produced bad offspring. Preferably, a rule is that if a file has one bad offspring then the inherited ability can be automatically removed.

f) An ability to watch clones or identical twins of objects (e.g. the same process running on other systems in the community) to compare the pattern of their issue and to

make decisions as to whether or not to treat any particular process as malware.

One or more of these features a) to f) can be used to provide a solution to the problem of producing a security product that can be used effectively without 100% reliance on being permanently connected to the Internet, which is often impractical. Examples of this are Windows Update and other processes used more and more by vendors who wish to be able to roll out product updates automatically across the worldwide web.

Possible benefits of these types of features above conventional software are as follows. Antivirus software tends to have a cache of known bad signatures. The problem is keeping this up to date. Take the simple example of someone buying a new computer. The computer comes with an antivirus product preloaded with a signature cache. Between the time when the PC was built, shipped to the store and bought by the user several days or weeks will have passed. The user starts the PC and is exposed to any new virus or malware which was created after the PC was built. Full protection requires the user to connect to the internet and download updates. This cannot be guaranteed to occur ahead of other activities by the user on the internet (almost physically impossible to guarantee). With a local cache of known good processes, as in the embodiments of the present invention, it is possible to ship the computer/terminal preloaded with a pre-generated cache of signatures for all of the good (clean) software preloaded by the computer manufacturer. In this case the user can connect to the internet knowing that any new or updated programs will be immediately detected and verified.

Also any auto-updating software can function forcing signatures to be automatically built for its children and more remote off-spring (i.e. grandchildren, great-grandchildren, etc).

Reference is now made to "Defeating Polymorphism: Beyond Emulation" by Adrian Stepan of Microsoft Corporation published in "Virus Bulletin Conference October 2005" and also to US 60/789156 filed on 5th April 2006, the entire content of which are hereby incorporated by reference. In that paper and patent application, there are disclosed methods of decrypting files to allow the files to be analysed to determine whether or not the file actually is malware. In US 60/789156 in particular, there is disclosed a method of unpacking/decrypting an executable computer file using a host computer, the method comprising: partitioning the executable computer file into plural basic blocks of code; translating at least some of the basic blocks of code into translated basic blocks of code that can be executed by the host computer; linking at least some of the translated basic blocks of code in memory of the host computer; and, executing at least some of the translated basic blocks of code on the host computer so as to enable the executable computer file to be unpacked or decrypted, whereupon the unpacked or decrypted executable computer file can be analyzed to determine whether the executable computer file is or should be classed as malware. There is also disclosed in US 60/789156 a method of unpacking/decrypting an executable computer file, the method comprising: partitioning the executable computer file into plural basic blocks of code; creating at least a read page of cache memory for at least some of the basic blocks, the read page of cache memory storing a read cached

real address corresponding to a read cached virtual memory address for the respective basic block, and creating at least a write page of cache memory for at least some of the basic blocks, the write page of cache memory storing a write cached real address corresponding to a write cached virtual memory address for the respective basic block; and, emulating the executable file by executing at least some of the basic blocks of code so as to enable the executable computer file to be unpacked or decrypted, whereupon the unpacked or decrypted executable computer file can be analyzed to determine whether the executable computer file is or should be classed as malware; wherein during the execution of a basic block, at least one of the read page and the write page of cache memory is checked for a cached real address corresponding to the virtual address that is being accessed for said basic block.

The techniques disclosed in these documents can be used in the present context when it is desired to analyse a file in detail. More generally however, the techniques disclosed in these papers, and particularly the enhanced techniques disclosed in US 60/789156, can be used to provide information about the activity of a file when it is run on a computer because the techniques disclosed in these papers emulate the running of the file and therefore allow the file's activity to be interpreted.

A further situation arises when users wish to install software while offline. In a preferred embodiment, when a user attempts to install new software while offline, agent software running on the user's computer 2 prompts the user for authorisation to allow the installation process to proceed such that the execution of the installation can

itself be "authorised" by the user. This authorisation by the user is treated as a "Genesis" event and will be so termed hereinafter. However there are some processes commonly used in installation of software that communicate with other existing programs on the installation machine, e.g. Microsoft's MSIEXEC.

The Genesis approach involves a process that generates signatures as a result of the user's authorisation on the user's computer 2. Those signatures are stored in the local database on the user's computer 2. In one embodiment, those locally stored signatures are referred to as necessary by the agent software during the installation process so that the installation can proceed. Alternatively, the security checks made by the agent software can be switched off during the installation process. The switching off may only be for a limited duration, such as a few minutes which should be sufficient to allow most software to be installed, the off time optionally being user-configurable.

In any event, once the installation has been completed and the user's computer 2 connected to the community database 7, the agent software on the user's computer 2 can upload the signatures relating to the installation from the user's local database to the community database 7. With corresponding data from other users' computers 2, the community database 7 can then be used to make a rapid determination that the installation of this particular software is benign.

As a variant to this embodiment, when a user's computer 2 is at some point in time on-line, the agent

software on the user's computer 2 may download signatures of a so-called "trusted installer" or "licensed installer". This allows the operation of a method such that a "licensed installer" and any child processes of the licensed installer are permitted to execute while a Genesis event is "current", e.g. within a period of minutes after an authorisation from the user. Preferably signatures of licensed installers are always downloaded, as and when added to the community database 7, to a remote computer 2 while online.

There may be further refinements to this method, such as to prevent licensed installers executing if a media change has occurred during the currency of a Genesis event. However, any unknown processes, which may be malware, may still be detected and blocked. Having a small number of licensed installers facilitates download, as minimal data is required, especially compared to downloading virus signature files. As another example, super-trusted installers, such as for example "Windows Update", may be employed whereby all new processes created by the super-trusted installer are marked immediately as safe.

In the case of an installation occurring when the remote computer 2 is connected to the community database 7, another option is that if the software that is about to be installed is not known to the community database 7, then the system will block the installation or alert the user. For example, a message may be displayed at the user's computer 2 to the effect that "You are about to install some software. This software is not known to [the community]. Are you sure you wish to proceed?".

Embodiments of the present invention have been described with particular reference to the examples illustrated. However, it will be appreciated that variations and modifications may be made to the examples described within the scope of the present invention.

CLAIMS

1. A method of classifying a computer object as malware, the method comprising:

at a base computer, receiving data about a computer object from each of plural remote computers on which the object or similar objects are stored;

comparing in the base computer the data about the computer object received from the plural computers; and,

classifying the computer object as malware on the basis of said comparison.

2. A method according to claim 1, wherein the data about the computer object that is sent from the plural remote computers to the base computer and that is used in the comparison includes one or more of: executable instructions contained within or constituted by the object; the size of the object; the current name of the object; the physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers.

3. A method according to claim 1, wherein the comparing identifies relationships between the object and other objects.

4. A method according to claim 3, wherein if at least one other object to which said object is related is classed as malware, then classifying said object as malware.

5. A method according to claim 3, wherein said other objects include the object or similar objects stored on at least some of the remote computers.

6. A method according to claim 3, wherein said other objects include other objects that are parent objects or child objects or otherwise process-related objects to said object.

7. A method according to claim 1, wherein the data is sent in the form of key that is obtained by a hashing process carried out in respect of the objects on the respective remote computers.

8. A method according to claim 7, wherein the key has at least one component that represents executable instructions contained within or constituted by the object.

9. A method according to claim 7, wherein the key has at least one component that represents data about said object.

10. A method according to claim 9, wherein said data about said object includes at least one of: the current name of the object; the physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the

object is created, configured or runs on the respective remote computers.

11. A method according to claim 7, wherein the key has at least one component that represents the physical size of the object.

12. A method according to claim 1, comprising initially classifying an object as not malware, generating a mask for said object that defines acceptable behaviour for the object, and comprising monitoring operation of the object on at least one of the remote computers and reclassifying the object as malware if the actual monitored behaviour extends beyond that permitted by the mask.

13. Apparatus for classifying a computer object as malware, the apparatus comprising:

a base computer constructed and arranged to receive data about a computer object from each of plural remote computers on which the object or similar objects are stored;

the base computer being constructed and arranged to compare the data about the computer object received from said plural computers; and,

the base computer being constructed and arranged to classify the computer object as malware on the basis of said comparison.

14. Apparatus according to claim 13, wherein the base computer is constructed and arranged so as to compare the data where the data includes one or more of: executable instructions contained within or constituted by the object; the size of the object; the current name of the object; the

physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers.

15. Apparatus according to claim 13, wherein the base computer is constructed and arranged so that the comparing identifies relationships between the object and other objects.

16. Apparatus according to claim 15, the base computer is constructed and arranged so that if at least one other object to which said object is related is classed as malware, then said object is classified as malware.

17. Apparatus according to claim 15, wherein the base computer is constructed and arranged so that said other objects include the object or similar objects stored on at least some of said remote computers.

18. Apparatus according to claim 15, wherein the base computer is constructed and arranged so that said other objects include other objects that are parent objects or child objects or otherwise process-related objects to said object.

19. Apparatus according to claim 13, wherein the base computer is constructed and arranged so as to be able to process data that is sent in the form of key that is

obtained by a hashing process carried out in respect of the objects on said respective remote computers.

20. Apparatus according to claim 19, wherein the key has at least one component that represents executable instructions contained within or constituted by the object.

21. Apparatus according to claim 19, wherein the key has at least one component that represents data about said object.

22. Apparatus according to claim 21, wherein said data about said object includes at least one of: the current name of the object; the physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers.

23. Apparatus according to claim 19, wherein the key has at least one component that represents the physical size of the object.

24. Apparatus according to claim 19, wherein the base computer is constructed and arranged to generate a mask for an object that is initially classed not as malware, said mask defining acceptable behaviour for the object, and the base computer is constructed and arranged to monitor operation of the object on at least one of the remote computers and to reclassify the object as malware if the

actual monitored behaviour extends beyond that permitted by the mask.

25. A method of providing data about a computer object from a remote computer to a base computer so that a comparison can be made at the base computer with similar data received from other remote computers, the method comprising:

providing from a remote computer to a base computer data about a computer object that is stored on the remote computer;

the data including one or more of: executable instructions contained within or constituted by the object; the size of the object; the current name of the object; the physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers;

the data being sent in the form of key that is obtained by a hashing process carried out in respect of the object on the remote computer.

26. A method according to claim 25, wherein the key has at least one component that represents executable instructions contained within or constituted by the object.

27. A method according to claim 25, wherein the key has at least one component that represents data about said object.

28. A method according to claim 27, wherein said data about said object includes at least one of: the current name of the object; the physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers.

29. A method according to claim 25, wherein the key has at least one component that represents the physical size of the object.

30. A computer program comprising program instructions for causing a computer to perform the method of claim 25.

31. A method of determining the protection that a remote computer has from malware, the method comprising:

receiving at a base computer details of all or selected security products operating at a point in time on said remote computer;

receiving similar information from other remote computers connected to the base computer; and,

identifying any malware processes that were not identified by said other remote computers having that particular combination of security products.

32. A method according to claim 31, comprising providing information to the user of said remote computer that said remote computer may be susceptible to attack by said malware processes on the basis of said identifying.

33. A method according to claim 31, wherein the details of all or selected security products includes the name of the security products, versions, and loaded signature files.

34. Apparatus for determining the protection that a remote computer has from malware, the apparatus comprising:

a base computer constructed and arranged to receive computer details of all or selected security products operating at a point in time on said remote computer;

the base computer being constructed and arranged to receive similar information from other remote computers connected to the base computer; and,

the base computer being constructed and arranged to identify any malware processes that were not identified by said other remote computers having that particular combination of security products.

35. Apparatus according to claim 34, wherein the base computer is constructed and arranged to provide information to the user of said remote computer that said remote computer may be susceptible to attack by said malware processes on the basis of said identifying.

36. Apparatus according to claim 34, wherein the details of all or selected security products includes the name of the security products, versions, and loaded signature files.

37. A method of classifying a computer object as malware or as safe, wherein said computer object is a descendant or otherwise related object of a first computer object, the method comprising:

classifying a first computer object as malware or as safe;

identifying in a key relating to said first computer object a component that uniquely identifies the first computer object and that is inherited or otherwise present in the key of a descendant or other related computer object of the first computer object; and,

classifying said computer object as malware or as safe as the case may be on the basis of the unique identifier component being present in the key of said computer object.

38. A method according to claim 37, comprising monitoring activities of said first computer object and reclassifying the first computer object as malware in the case that it was initially classified as safe and subsequently determined to be malware, the method further comprising automatically classifying as malware any computer object that has a key in which said unique identifier component is present.

39. A computer program comprising program instructions for causing a computer to perform the method of claim 38.

40. Apparatus for classifying a computer object as malware or as safe, wherein said computer object is a descendant or otherwise related object of a first computer object, the apparatus comprising:

a computer constructed and arranged to classify a first computer object as malware or as safe;

the computer being constructed and arranged to identify in a key relating to said first computer object a component that uniquely identifies the first computer object and that is inherited or otherwise present in the

key of a descendant or other related computer object of the first computer object; and,

the computer being constructed and arranged to classify said computer object as malware or as safe as the case may be on the basis of the unique identifier component being present in the key of said computer object.

41. Apparatus according to claim 40, wherein the computer is constructed and arranged to monitor activities of said first computer object and reclassify the first computer object as malware in the case that it was initially classified as safe and subsequently determined to be malware, the computer further being constructed and arranged to automatically classify as malware any computer object that has a key in which said unique identifier component is present.

42. A method of installing software on a computer, the method comprising:

on initiation of installation of software on a computer, providing a computer-generated prompt on the computer to a user to ascertain whether the user authorises the installation; and,

ceasing the installation if a user authorisation is not received, else:

receiving at the computer the user's authorisation to proceed with the installation;

proceeding with the installation;

obtaining data about computer objects that are created or used during the installation;

storing said data at the local computer.

43. A method according to claim 42, wherein the locally stored data is referred to during the installation to ensure that all objects created or used during the installation are part of the installation process, and, if it is found that objects created or used during the installation are not part of the installation process, either or both of: (i) ceasing the installation and (ii) providing a computer-generated prompt on the computer to the user accordingly.

44. A method according to claim 42, comprising connecting the computer to a community database that is connectable to a plurality of computers, and uploading the stored data to the community database for comparison with similar data provided by other computers.

45. A method according to claim 42, comprising downloading data about trusted installers to the computer, said data about trusted installers being referred to during the installation such that any objects relating to or created by the trusted installer are automatically authorised to proceed.

46. A method according to claim 45, wherein said data about trusted installers is referred to only for a predetermined time period following receipt at the computer of the user's authorisation to proceed with the installation.

47. A computer program comprising program instructions for causing a computer to perform the method of claim 42.

METHODS AND APPARATUS FOR DEALING WITH MALWARE

ABSTRACT

In one aspect, a method of classifying a computer object as malware includes receiving at a base computer data about a computer object from each of plural remote computers on which the object or similar objects are stored. The data about the computer object received from the plural computers is compared in the base computer. The computer object is classified as malware on the basis of said comparison. In one embodiment, the data about the computer object includes one or more of: executable instructions contained within or constituted by the object; the size of the object; the name of the object; the logical storage location or path of the object on the respective remote computers; the vendor of the object; the software product and version associated with the object; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers.

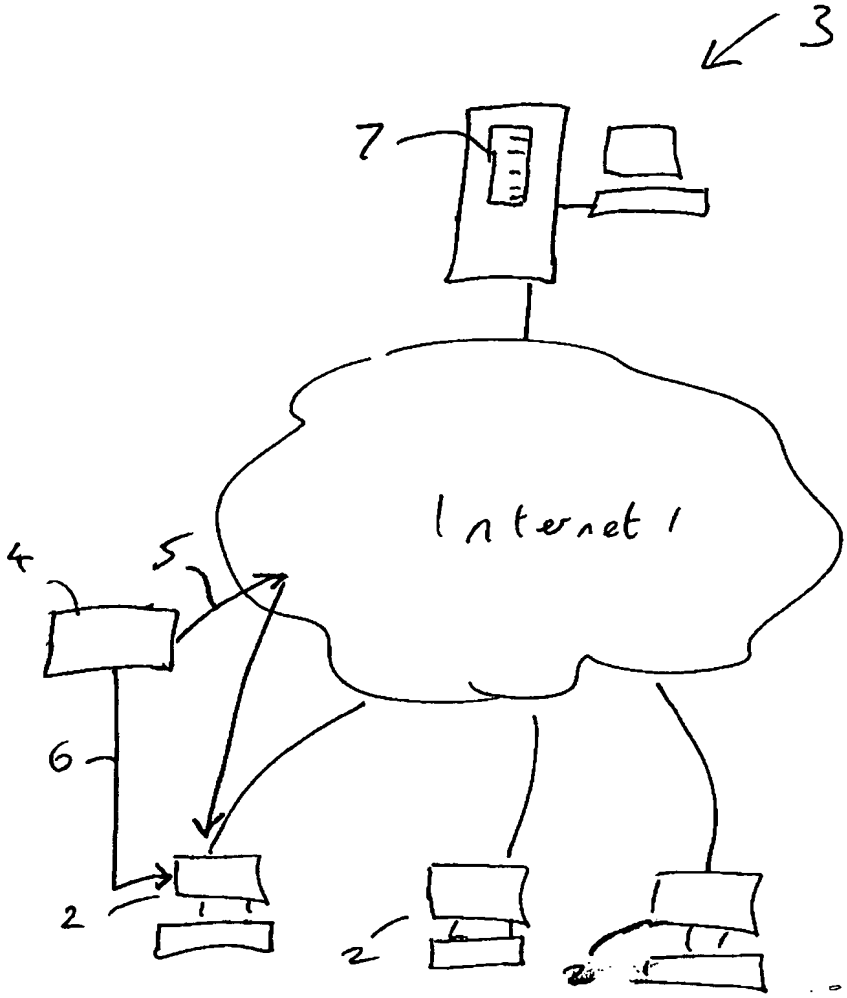


FIG. 1

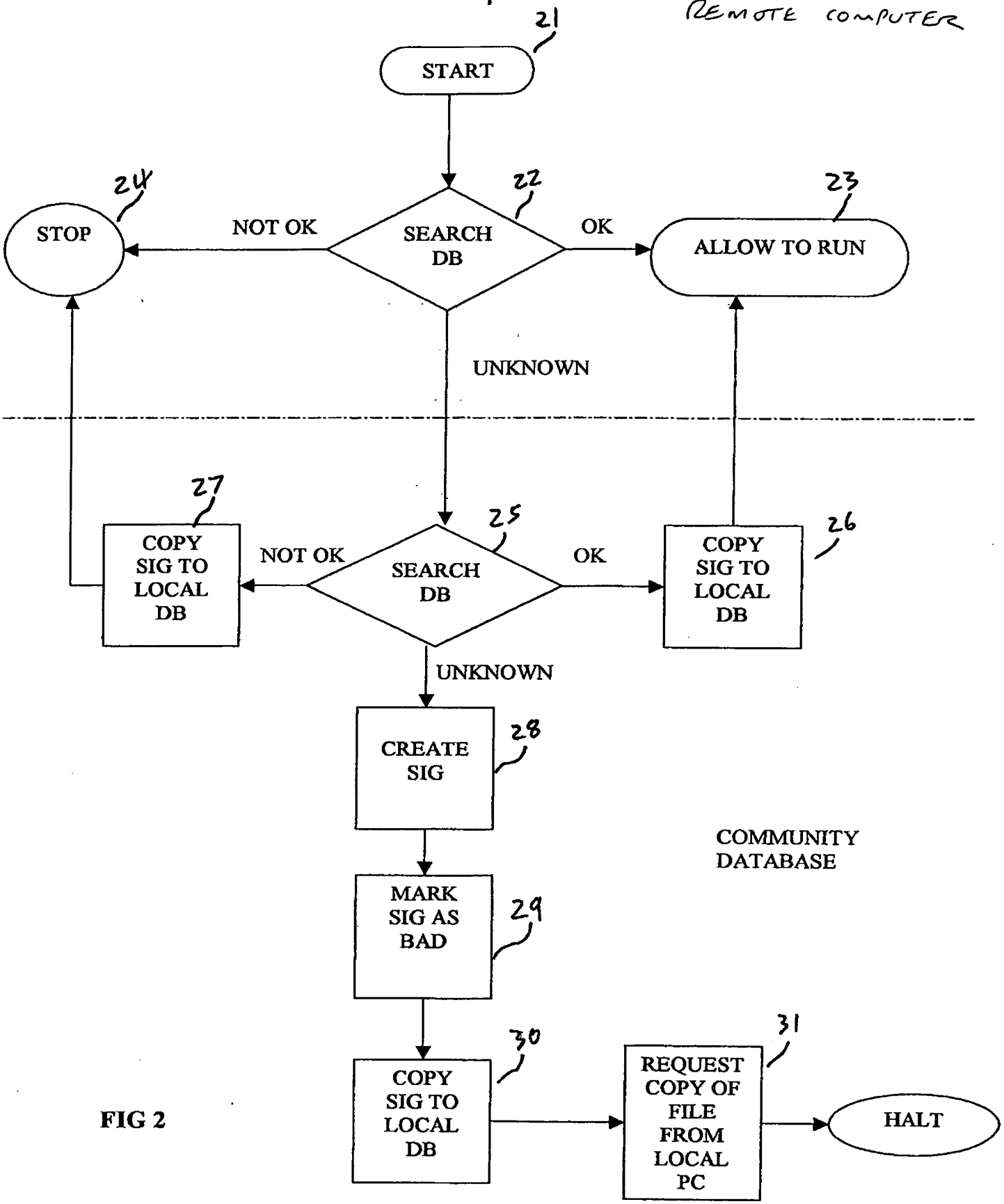

REMOTE COMPUTER
21
START
22
SEARCH DB
NOT OK
OK
24
STOP
23
ALLOW TO RUN
UNKNOWN
27
COPY SIG TO LOCAL DB
25
SEARCH DB
NOT OK
OK
26
COPY SIG TO LOCAL DB
UNKNOWN
28
CREATE SIG
COMMUNITY DATABASE
29
MARK SIG AS BAD
30
COPY SIG TO LOCAL DB
31
REQUEST COPY OF FILE FROM LOCAL PC
HALT

FIG 2

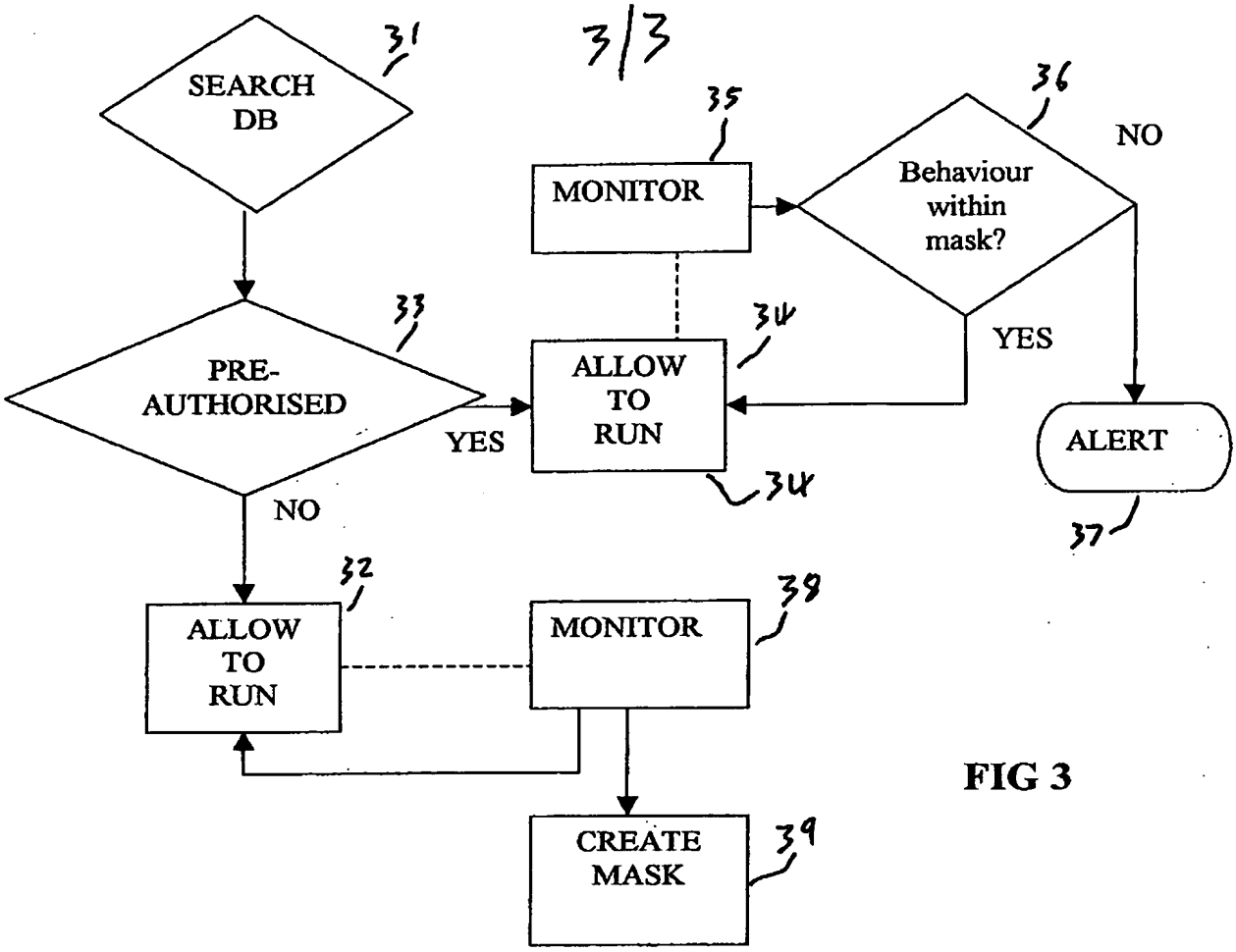

SEARCH DB
31
PRE-AUTHORISED
33
YES
NO
ALLOW TO RUN
32
MONITOR
35
Behaviour within mask?
36
NO
YES
ALLOW TO RUN
34
ALERT
37
MONITOR
38
CREATE MASK
39

FIG 3

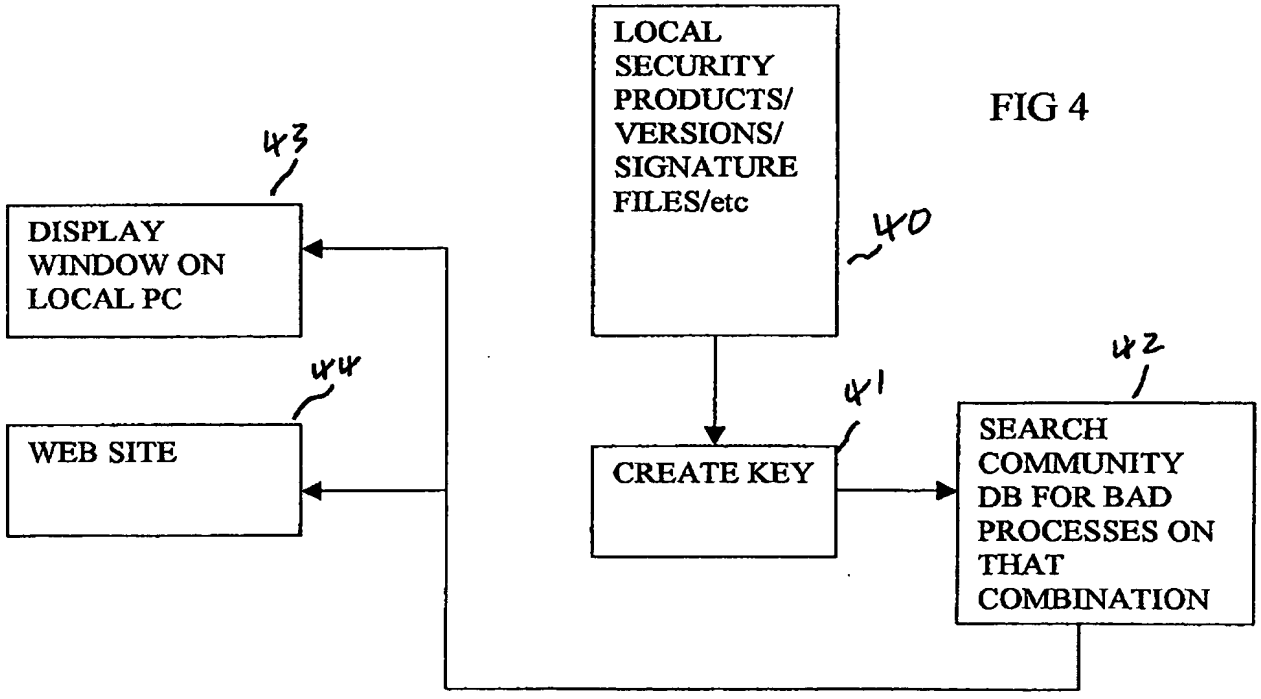

LOCAL SECURITY PRODUCTS/ VERSIONS/ SIGNATURE FILES/etc
40
DISPLAY WINDOW ON LOCAL PC
43
WEB SITE
44
CREATE KEY
41
SEARCH COMMUNITY DB FOR BAD PROCESSES ON THAT COMBINATION
42

FIG 4

Attorney Docket No. 011765-0355986 *PATENT*
Client Reference: AF/JG/P10717US

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: MELVYN MORRIS, PAUL STUBBS, MARKUS HARTWIG and DARREN HARTER

Application No.:	NEW	Confirmation No:	Unknown
Filed:	June 30, 2006	Group No.:	TBA
		Examiner	Unknown

For: METHODS AND APPARATUS FOR DEALING WITH MALWARE

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPLICATION DATA SHEET
37 C.F.R. § 1.76

BIBLIOGRAPHIC DATA

1. Applicant information

First applicant: MELVYN MORRIS
Citizenship: Great Britain
Residence: Redmire Gap, Intake Lane, Turnditch, Derbyshire DE56 2LU, United Kingdom

Second applicant: PAUL STUBBS
Citizenship: Great Britain
Residence: 7 Rookery Road, Wyboston, Bedfordshire MK44 3AX, United Kingdom

Third applicant: MARKUS HARTWIG
Citizenship: Germany
Residence: Exchange House, 494 Midsummer Boulevard, Milton Keynes, Buckinghamshire MK9 2EA, United Kingdom

Fourth applicant: DARREN HARTER
Citizenship: Great Britain
Residence: 178 Barnwood Road, Hucclecote, Gloucester, Gloucestershire GL4 3JZ, United Kingdom

2. Correspondence information

Correspondence for this application should be addressed as follows:

Customer No.: 00909

3. Application information

Title of Invention: METHODS AND APPARATUS FOR DEALING WITH MALWARE

Docket number assigned to this application: 011765-0355986
Suggested Classification: Class:
Subclass:
Technology Center to which subject matter is assigned:

Total number of drawing sheets: 3
Type of application: Utility
Application is to be published. Suggested drawing figure for publication: Figure 1

Secrecy order under § 5.2: This application does not disclose subject matter of an application which is under a secrecy order pursuant to § 5.2.

4. Representative information

The following have a power of attorney or authorization of agent in this application:

Customer No.: 00909

5. Foreign Priority information

Foreign priority for this application is claimed as follows:

35 U.S.C. § 365(c):

Application No.: GB 0513375.6
Filed: June 30, 2005
Status: Pending

6. Assignee information

The assignee(s) of this application is/are:

PREVX LIMITED
3 New Park Place
Pride Park
Derby DE24 8DZ
United Kingdom

Extent of interest of assignee in application: Entire right, title and interest.

Date: June 30, 2006

Christine M. McCarthy
Registration No. 41844
PILLSBURY WINTHROP SHAW PITTMAN LLP
P.O. Box 10500
McLean, VA 22102
(703.) 770.7743
Customer No. 00909

PTO/SB/06 (12-04)
Approved for use through 7/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875	Application or Docket Number 11,477,807

APPLICATION AS FILED – PART I

	(Column 1)	(Column 2)	SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
FOR	NUMBER FILED	NUMBER EXTRA	RATE ($)	FEE ($)		RATE ($)	FEE ($)
BASIC FEE (37 CFR 1.16(a), (b), or (c))				150			300
SEARCH FEE (37 CFR 1.16(k), (i), or (m))				250			500
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))				100			200
TOTAL CLAIMS (37 CFR 1.16(i))	47 minus 20 =	27	X$ 25		OR	X$50	1350
INDEPENDENT CLAIMS (37 CFR 1.16(h))	8 minus 3 =	5	X$100			X$200	1000
APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR						
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))			180			360	
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL			TOTAL	3350

APPLICATION AS AMENDED – PART II

		(Column 1)		(Column 2)	(Column 3)	SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE ($)	ADDI-TIONAL FEE ($)		RATE ($)	ADDI- TIONAL FEE ($)
AMENDMENT A	Total (37 CFR 1.16(i))	*	Minus	**	=	x =		OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =		OR	x =	
	Application Size Fee (37 CFR 1.16(s))									
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))					180		OR	360	
						TOTAL ADD'T FEE		OR	TOTAL ADD'T FEE	

		(Column 1)		(Column 2)	(Column 3)			OR		
		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE ($)	ADDI-TIONAL FEE ($)		RATE ($)	ADDI- TIONAL FEE ($)
AMENDMENT B	Total (37 CFR 1.16(i))	*	Minus	**	=	x =		OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =		OR	x =	
	Application Size Fee (37 CFR 1.16(s))									
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))					N/A		OR	N/A	
						TOTAL ADD'T FEE		OR	TOTAL ADD'T FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/SB/06 (12-04)
Approved for use through 7/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD
Substitute for Form PTO-875

Application or Docket Number: 10/647,807

APPLICATION AS FILED – PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	SMALL ENTITY RATE ($)	SMALL ENTITY FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY FEE ($)
BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A			N/A	300
SEARCH FEE (37 CFR 1.16(k), (l), or (m))	N/A	N/A	N/A			N/A	500
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	200
TOTAL CLAIMS (37 CFR 1.16(i))	9 minus 20 =		x =		OR	x =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	1 minus 3 =		x =			x =	
APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).						
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))			N/A			N/A	
			TOTAL			TOTAL	1,000

* If the difference in column 1 is less than zero, enter "0" in column 2.

APPLICATION AS AMENDED – PART II

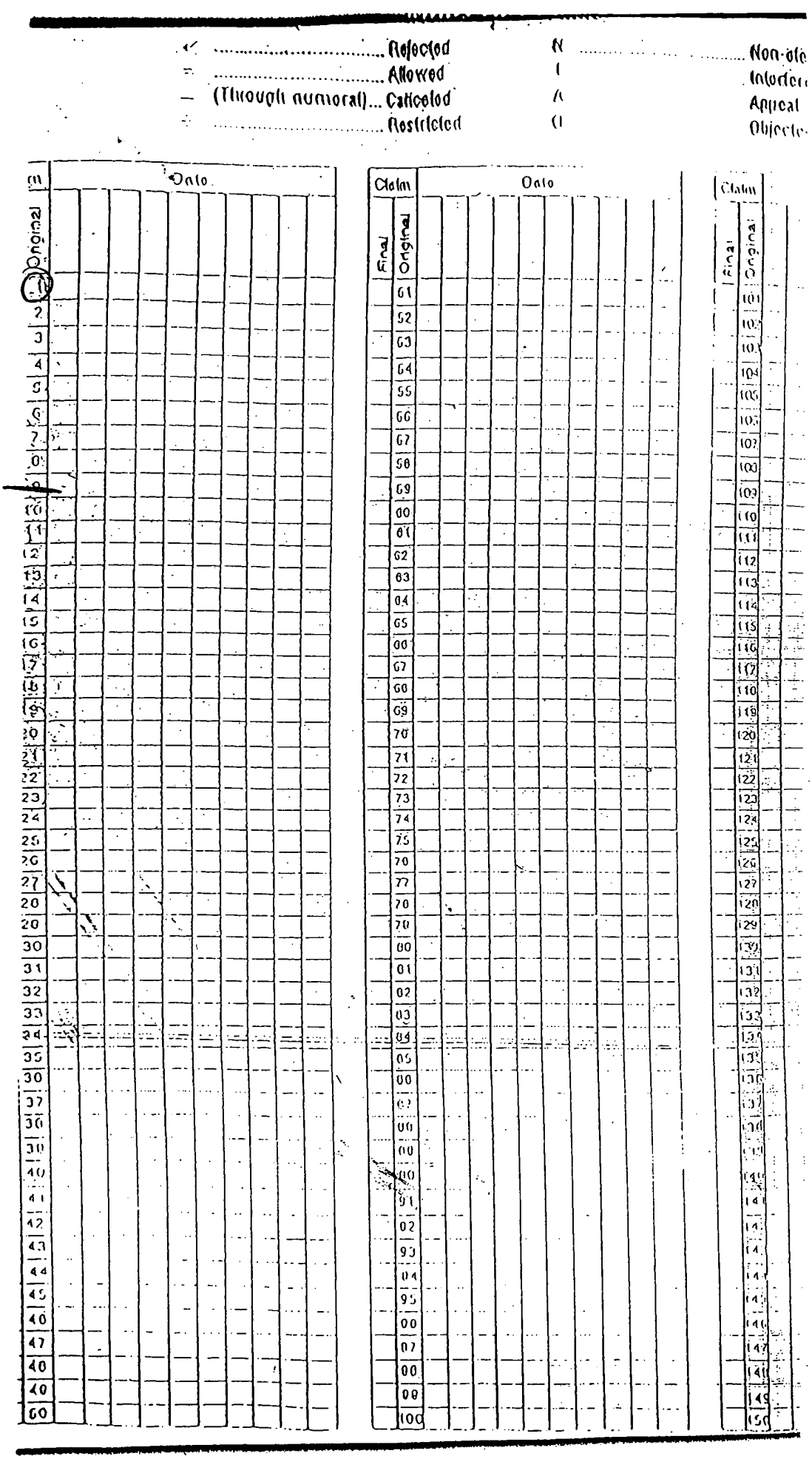
		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE ($)	SMALL ENTITY ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE ($)
AMENDMENT A	Total (37 CFR 1.16(i))	* 9	Minus	** 20	=	x =		OR	x =	
	Independent (37 CFR 1.16(h))	* 1	Minus	*** 3	=	x =		OR	x =	
	Application Size Fee (37 CFR 1.16(s))									
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))					N/A		OR	N/A	
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)	OR	RATE ($)	ADDITIONAL FEE ($)
AMENDMENT B	Total (37 CFR 1.16(i))	*	Minus	**	=	x =		OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =		OR	x =	
	Application Size Fee (37 CFR 1.16(s))									
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))					N/A		OR	N/A	
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to comp including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comm on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. P and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.


R ..... Rejected
A ..... Allowed
C (Through numeral) ... Cancelled
S ..... Restricted
N ..... Non-ele
I ..... Interfer
A ..... Appeal
O ..... Objecte
Date
Claim
Original
Final
Date
Claim
Original
Final