(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.

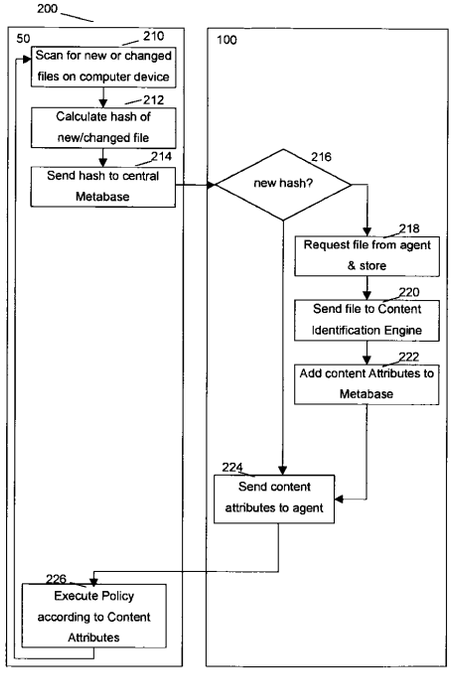

Fig. 3

EP 1 549 012 A1

IPR2023-01459
Trend Micro, Inc. EX1015 Page 1

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-

IPR2023-01459
Trend Micro, Inc. EX1015 Page 2

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

IPR2023-01459
Trend Micro, Inc. EX1015 Page 3

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-

IPR2023-01459
Trend Micro, Inc. EX1015 Page 4

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",

IPR2023-01459
Trend Micro, Inc. EX1015 Page 5

"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

IPR2023-01459
Trend Micro, Inc. EX1015 Page 6

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

IPR2023-01459
Trend Micro, Inc. EX1015 Page 7

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

IPR2023-01459
Trend Micro, Inc. EX1015 Page 8

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

IPR2023-01459
Trend Micro, Inc. EX1015 Page 9

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

IPR2023-01459
Trend Micro, Inc. EX1015 Page 10

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

IPR2023-01459
Trend Micro, Inc. EX1015 Page 11

- 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

IPR2023-01459
Trend Micro, Inc. EX1015 Page 12

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.

IPR2023-01459
Trend Micro, Inc. EX1015 Page 13

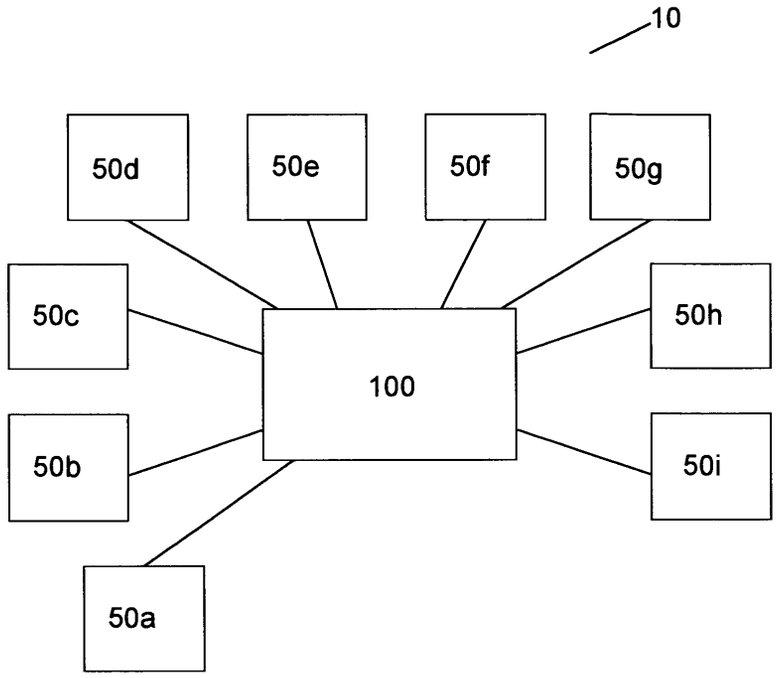
10
50d
50e
50f
50g
50c
50h
100
50b
50i
50a

Fig. 1

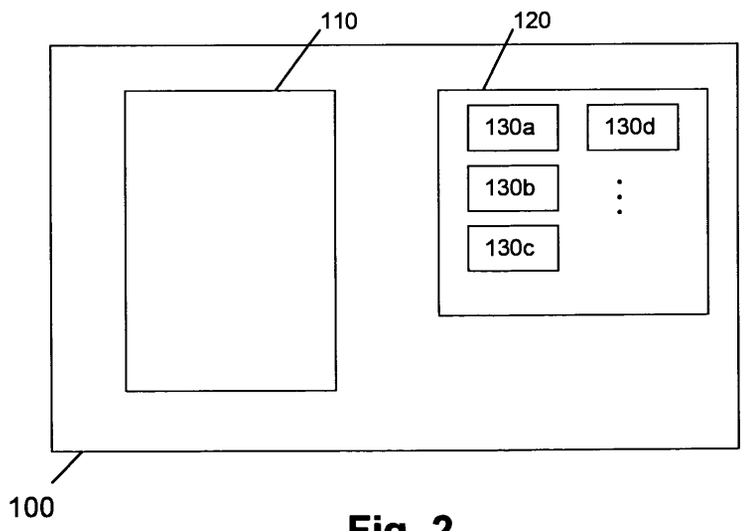
110
120
130a
130d
130b
130c
100

Fig. 2

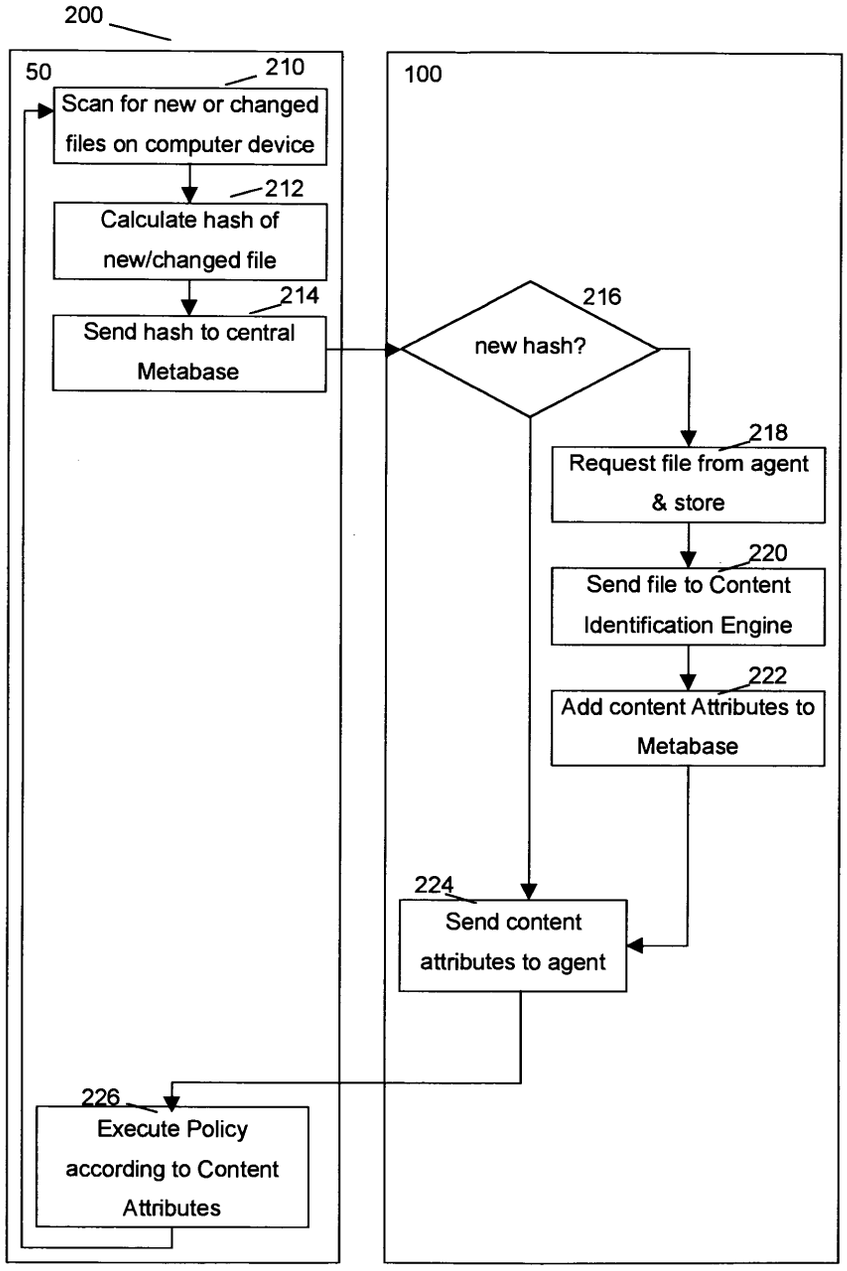
200
50
210
Scan for new or changed files on computer device
212
Calculate hash of new/changed file
214
Send hash to central Metabase
100
216
new hash?
218
Request file from agent & store
220
Send file to Content Identification Engine
222
Add content Attributes to Metabase
224
Send content attributes to agent
226
Execute Policy according to Content Attributes

Fig. 3

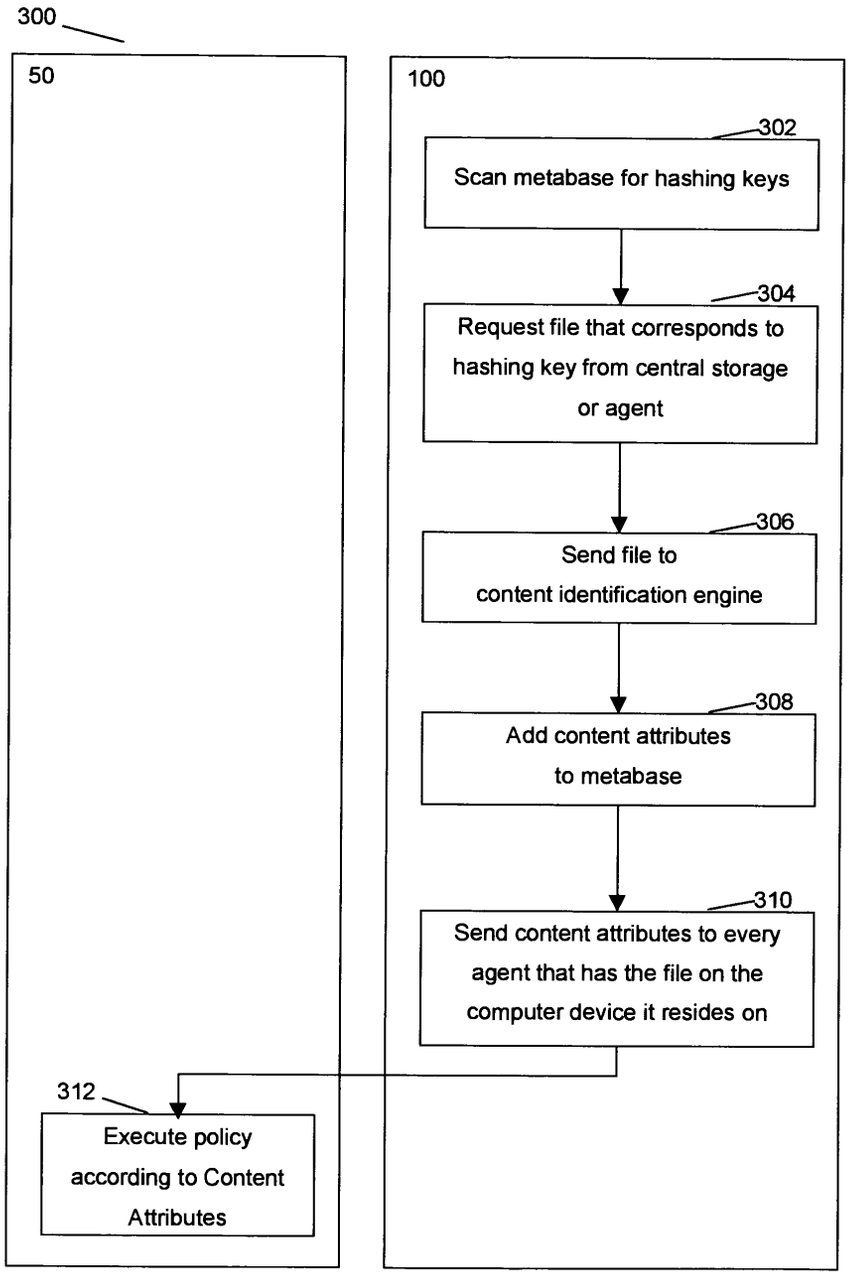
300
50
100
302
Scan metabase for hashing keys
304
Request file that corresponds to hashing key from central storage or agent
306
Send file to content identification engine
308
Add content attributes to metabase
310
Send content attributes to every agent that has the file on the computer device it resides on
312
Execute policy according to Content Attributes

Fig. 4

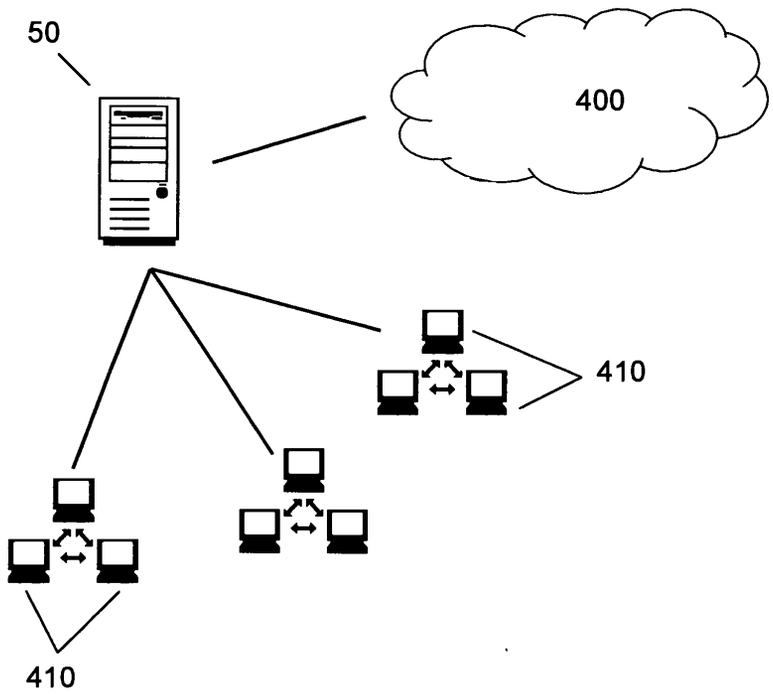

Fig. 5

IPR2023-01459
Trend Micro, Inc. EX1015 Page 17

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

1

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)

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

IPR2023-01459
Trend Micro, Inc. EX1015 Page 19