30. The method according to claim 23, wherein the transmitted location signal may contain a plurality of data.

31. The method according to claim 30, wherein the plurality of data are location coordinates derived from a Global Positioning System.

32. The method according to claim 23, wherein the transmitted location signal contains a "time stamp".

33. The method according to claim 23, wherein computer access is granted if the location of the location device matches the authorized remote log-in location.

34. The method according to claim 23, wherein computer access is denied if the location of the location device does not match the authorized remote log-in location.

35. An apparatus for enhancing computer security using a location device, comprising:

a central computer;

means for receiving a location signal sent from the location device to the central computer, the location signal containing at least a location of the location device,

wherein access of the central computer is determined on the basis of the location of the location device matching a pre-registered access location.

36. The apparatus according to claim 35, further comprising:

means for determining the location of the locating device.

37. The apparatus according to claim 36, further comprising:

a means for communicating between the central computer and a remote station.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 903

38. An apparatus for enhancing computer security using a location device, comprising:

a memory storing at least one authorized remote log-in location information;

a means for allowing a remote log-in contact;

an activator activating transmission of a location signal from the location device;

a receiver receiving the transmission of the location signal;

a means for determining a location of the location device on the basis of the transmitted location signal;

a central computer which determines whether the location of the location device is an authorized location with reference to the stored at least one authorized remote log-in location information.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 904

FIG. 1

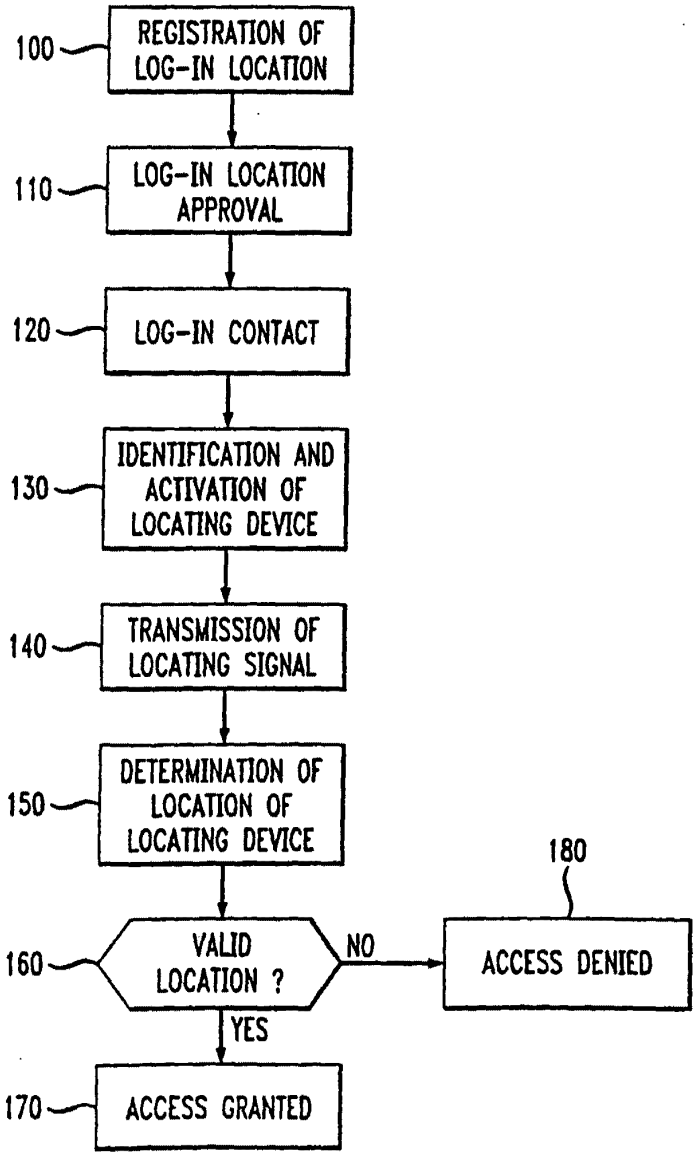
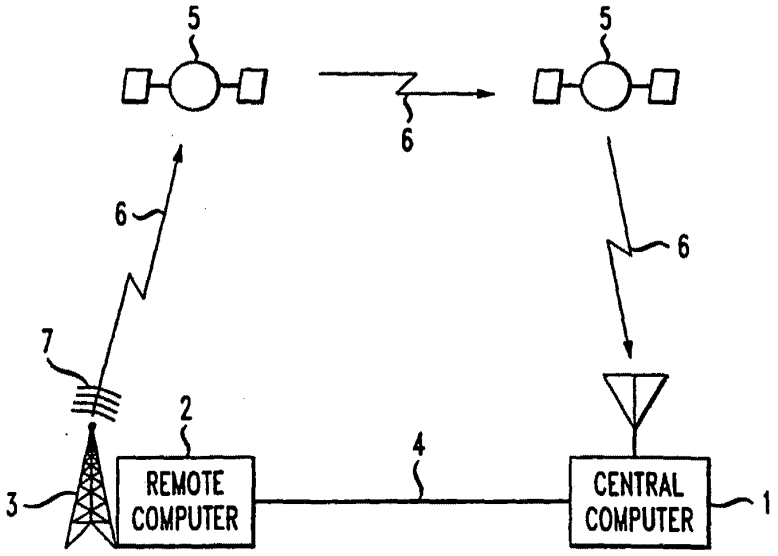

100
REGISTRATION OF LOG-IN LOCATION
110
LOG-IN LOCATION APPROVAL
120
LOG-IN CONTACT
130
IDENTIFICATION AND ACTIVATION OF LOCATING DEVICE
140
TRANSMISSION OF LOCATING SIGNAL
150
DETERMINATION OF LOCATION OF LOCATING DEVICE
160
VALID LOCATION ?
NO
180
ACCESS DENIED
YES
170
ACCESS GRANTED

FIG. 2



INTERNATIONAL SEARCH REPORT

Inter nal Application No
PCT/US 99/05025

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 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 6 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)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 97 13341 A (INTERNATIONAL SERIES RESEARCH INC.) 10 April 1997 see page 10, line 29 - page 12, line 18 see page 15, line 12 - line 32 see page 45, line 13 - line 31; figures 1,3A,3B,4-6 ---	1-3, 6-12,14, 17-24, 29-38
X	US 4 962 449 A (SCHLESINGER) 9 October 1990 see column 3, line 11 - line 34 see column 3, line 46 - column 4, line 18; figure 1 --- -/--	1-3,5, 10-12, 14,16, 21-24, 26,28, 33-38

[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
5 July 1999	12/07/1999

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	Taylor, P

2

Form PCT/ISA/210 (second sheet) (July 1992)

IPR2024-00027
CrowdStrike Exhibit 1002 Page 907

INTERNATIONAL SEARCH REPORT

Inte: ›nal Application No
PCT/US 99/05025

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
A	WO 93 21581 A (SECURE COMPUTING CORP.) 28 October 1993 see page 7, line 31 - page 8, line 23 see page 40, line 32 - page 43, line 25; figures 29-33 -----	1-3,5-7, 10-12, 14, 16-18, 21-24, 26, 28-30, 33-38

2

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

IPR2024-00027
CrowdStrike Exhibit 1002 Page 908

INTERNATIONAL SEARCH REPORT

Information on patent family members

Inte onal Application No
PCT/US 99/05025

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9713341 A	10-04-1997	US 5757916 A	26-05-1998
		AU 7392196 A	28-04-1997
		CA 2233962 A	10-04-1997
		EP 0880839 A	02-12-1998
US 4962449 A	09-10-1990	NONE	
WO 9321581 A	28-10-1993	US 5276735 A	04-01-1994
		AT 154150 T	15-06-1997
		AU 4284793 A	18-11-1993
		AU 678937 B	12-06-1997
		AU 5081196 A	18-07-1996
		CA 2118246 A	28-10-1993
		DE 69311331 D	10-07-1997
		DE 69311331 T	30-10-1997
		DK 636259 T	07-07-1997
		EP 0636259 A	01-02-1995
		EP 0737907 A	16-10-1996
		JP 7505970 T	29-06-1995
		US 5502766 A	26-03-1996
		US 5499297 A	12-03-1996

Form PCT/ISA/210 (patent family annex) (July 1992)

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 11/00, 11/22 — A1

(11) International Publication Number: **WO 98/41919**

(43) International Publication Date: 24 September 1998 (24.09.98)

(21) International Application Number: PCT/US98/03796

(22) International Filing Date: 13 March 1998 (13.03.98)

(30) Priority Data:
08/820,649 18 March 1997 (18.03.97) US

(71) Applicant: TREND MICRO, INCORPORATED [CN/–]; SF No. 218, Chin-shan S. Road, Sec. 2, Taipei (TW).

(71)(72) Applicant and Inventor: CHEN, Eva [CN/US]; 10408 Orange Avenue, Cupertino, CA 95014 (US).

(72) Inventors: LAU, Steven, Yuen-Lam; 2275 Truckee Drive, Santa Rosa, CA 95401 (US). LIANG, Yung-Chang; 820 Humerick Way, Sunnyvale, CA 94087 (US).

(74) Agents: KLIVANS, Norman, R. et al.; Skjerven, Morrill, MacPherson, Franklin & Friel LLP, Suite 700, 25 Metro Drive, San Jose, CA 95110 (US).

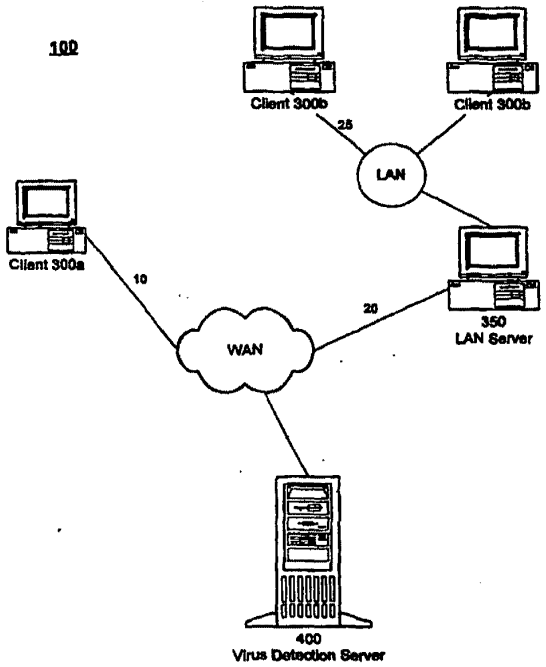
(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, GW, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) Title: VIRUS DETECTION IN CLIENT–SERVER SYSTEM

(57) Abstract

The iterative detection and treatment of viruses using virus detection objects and virus treatment objects is disclosed. Pursuant to a request for a virus scan, a virus detection object is produced by a server (400) and is transmitted to a client (300A, B) for execution. The client (300A, B) receives and executes the virus detection object, and the results are transmitted to the server (400). The server (400) uses the results to produce an additional virus detection object which is also transmitted to the client (300A, B) and executed so that the results can be transmitted to the server (400). The iterative production and execution of virus detection objects is continued until a determination is made as to whether the targeted file or data includes a virus. Upon a determination that a targeted file or data includes a virus, a vaccine specifically tailored to the conditions presented at the client (300A, B) and the type of virus detected is produced, preferably in the form of a virus treatment object. The request for a virus scan can be directly made or indirectly by a triggering event. One virus detection server (400) embodiment includes a virus information expert system that applies conditional data to predetermined knowledge about virus scanning to make determinations such as when to scan for viruses. A network diagnosis and treatment application includes a diagnostic data module, an expert system, and a maintenance requesting module. The expert system applies the diagnostic data to predetermined knowledge about the diagnosis and maintenance of a network to make determinations upon which maintenance requests are made. Preferably, the expert system includes a virus information expert system.



IPR2024-00027
CrowdStrike Exhibit 1002 Page 910

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

VIRUS DETECTION IN CLIENT-SERVER SYSTEM

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates generally to the detection of viruses in computer files and more particularly to the detection and treatment of viruses using network resources, the iterative detection and treatment of viruses, and the maintenance of computer networks.

Description of the Related Art

Computer viruses continue to be problematic to computers and computer users. Such viruses are typically found within computer programs, files, or code and can produce unintended and sometimes damaging results. One type of virus is a portion of program code that produces copies of itself in other programs, allows the programs to perform their regular operations, and surreptitiously performs other, unintended actions. Other types of viruses include, without limitation, the following: worms, logic bombs, time bombs, trojan horses, and any malicious program or code residing in executable programs, macros, applets, or elsewhere. While advances have been made in the detection of viruses, the proliferation of computers and the increasing interconnection of, and communication between, computers have also increased the opportunities for the spread of existing viruses and the development of new computer viruses. Thus, the number and type of viruses to which a computer or computer system is potentially exposed is ever changing: This is one reason that the information used to detect viruses requires seemingly constant revision and augmentation in order to detect the various strains of viruses.

There are various methods for detecting viruses. One method of detection is to compare known virus signatures to targeted files to determine whether the targeted files include a virus signature and, thus, the corresponding virus. The

IPR2024-00027
CrowdStrike Exhibit 1002 Page 912

comparison data used for virus detection might include a set of such known virus signatures and, possibly, additional data for virus detection. Typically, the comparison data is maintained in a computer storage medium for access and use in the detection of viruses. For example, for a personal computer the comparison data might be stored on the computer's hard disk. Periodically, comparison data updates are provided to detect new or different forms of viruses. The comparison data updates are typically provided on some source storage medium for transfer to the storage medium used to maintain the comparison data. For example, an update might be provided on a floppy disk so that a personal computer user can transfer the comparison data update from the floppy disk to the computer hard disk to complete the update.

The comparison data is essentially discrete and static. That is, all of the information used for the detection of viruses generally remains constant unless it is updated or altered by the user or other relevant party or action. This can be problematic because the quality of information used to detect viruses is reliant upon some form of comparison data maintenance. Another problem with updatable comparison data is that the comparison data can quickly lose its efficacy due to the existence of new and different viruses. Thus, while a periodic update might seem effective, there is no telling how many new and different viruses could be produced in the interim. Still another problem with comparison data updates is that a transfer of an entire replacement set of data, or at least a transfer of all the new virus detection data, is typically undertaken in order to complete the update. Whether an entire replacement or all of the new virus detection data is involved, a significant amount of data must be transferred for the update. More specifically, if a user updates her virus detection information using, for example, an update provided on a floppy disk, at least all of the new virus detection information is transferred from the floppy disk to the appropriate medium.

Regardless of the update source, the problems of updatable comparison data remain. Specifically, the user, administrator, or other relevant party is still

IPR2024-00027
CrowdStrike Exhibit 1002 Page 913

typically responsible for accessing and updating the comparison data, the comparison data can quickly and unpredictably lose its efficacy, and a significant amount of data must be transferred from the source to the storage medium used for the comparison data. Indeed, the amount of data to be transferred may be more problematic where internet resources are the source of the comparison data update since a significant amount of computational resources would be used to complete the update.

Another problem in the detection of viruses is that conditions vary from computer to computer. Thus, a first computer or medium could require a first type of scanning while another computer or medium, even one in the same network as the first, could require a second type of scanning. In these instances, virus scans can be overinclusive in that the scanning for viruses that could not possibly reside at the computer, and can be underinclusive if an exhaustive scan for the types of viruses likely to reside at the computer, based upon the conditions presented at the computer, is not undertaken. To adequately perform a virus scan according to the conditions particular to a computer, a user or other relevant party typically must configure the scan. This can be problematic because of reliance upon party input. Additionally, the conditions pertaining to a particular computer and the requisite type of scanning can change.

With the increasing interconnection and communication between computers, the requirements for maintaining computers residing on a computer network have also increased. Again, maintenance is typically undertaken directly by a person, such as the network administrator, using resources which are locally available to the network administrator. For example, in the treatment of computers on a local area network for viruses, an administrator could commonly configure the computers to access locally available virus scanning resources. This maintenance scheme is problematic in its reliance upon updates, its failure to adapt to changing conditions, and its failure to make adequate use of resources external to the local area network.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 914

Accordingly, there remains a need for virus detection that obviates some of the problems inherent in the updatable comparison data model. Additionally, there remains a need for more effective implementation of internet resources for virus detection and for easier and more effective computer network maintenance, particularly, by way of example, with respect to virus detection and elimination.

SUMMARY OF THE INVENTION

In accordance with the present invention a virus detection server is provided for centralized access and iterative techniques are provided to effectively detect and treat viruses.

In one aspect of the present invention, viruses are iteratively detected at a client computer. A substantial portion of the tools and information required for the detection and treatment of viruses is provided in a centralized location such as a server, preferably an internet or world wide web server. This virus detection server operates in conjunction with a client to determine whether viruses reside at the client. A virus scan is initiated when a request is received or directed at the virus detection server. The request is direct or can be initiated by various triggering events, such as a programmed request from the client that does not require ongoing user initiation such that the scan is initiated without a request that it apparent to the user.

Once it is determined by the virus detection server that a valid virus detection request has been received, the virus detection server operates to iteratively detect and treat viruses associated with the requester, typically the client. The iterative production of virus detection objects allows objects to be specifically tailored according to previously determined conditions and/or conditions discovered as a result of the execution of previously produced virus detection objects. Specifically, a virus detection object is produced by the virus detection server and is transmitted to the client. The virus detection object includes an executable program which the client includes a corresponding executing engine.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 915

Thus, when the client receives the virus detection object, it executes the object and produces a result that is transmitted back to the virus detection server. The results of the execution of the virus detection object are transmitted to the virus detection server so that the server can produce additional virus detection objects based upon the results of the execution of the previous virus detection object or objects.

According to another aspect of the present invention, the virus detection server produces a vaccine specific to the conditions presented at the client, including the specific type of virus that was detected. The virus detection server receives the results produced by the execution of one or more virus detection objects and uses the results to tailor the vaccine. For example, the result may indicate that a specific virus is present in a specific type of file on a specific platform, and the vaccine is tailored accordingly. The vaccine comprises one or more virus treatment object which include an executable program and/or a data set. The virus treatment objects are sent to and executed at the client to treat the viruses.

According to still another aspect of the present invention, a system of computers such as those residing on a local area network can be effectively screened for viruses without requiring constant maintenance on the part of a system administrator. Specifically, events that regularly occur can be used to trigger a request for virus scanning by the virus detection server.

An embodiment of the virus detection server includes various modules for the iterative detection and treatment of computer viruses. An iterative virus detection module includes a scanning module, a virus pattern module, a virus rules module, a cleaning module, a cleaning pattern module, an access managing module and an access data module. The scanning module includes a plurality of routines for detecting viruses which are provided such that they can be separately accessed. Similarly, the virus pattern module and the virus rules module respectively include groups of virus signatures and groups of rules that can be separately accessed. The iterative virus detection module operates in conjunction with the scanning module,

IPR2024-00027
CrowdStrike Exhibit 1002 Page 916

the virus pattern module, and the virus rules module to produce virus detection objects that are specifically provided for the detection of viruses based upon the conditions presented at the client.

The cleaning module includes routines for treating files that are determined to include a virus. A plurality of routines that can be separately accessed are provided in the cleaning module, as are a plurality of virus detection patterns in the virus detection module. The iterative virus detection module operates in conjunction with the cleaning module and the cleaning pattern module to produce virus treatment objects that are tailored for the treatment of the particular virus or viruses that are detected at the client in the prevailing conditions presented at the client.

The access managing module includes routines for determining whether a request for virus scanning has been provided and for validating the request upon such a determination. The access managing module operates in conjunction with the access data module, which includes information pertaining to potential users, in order to acknowledge and validate virus scanning requests.

According to still another aspect of the present invention, another embodiment of a virus detection server includes a virus information expert system. Conditional data pertaining to the conditions presented at computers, media, or other scanning targets is gathered and maintained. The conditional data is applied to predetermined knowledge to make determinations used in conducting a virus scan. An embodiment of the expert system includes an inference engine and a virus knowledge module. The inference engine accesses the conditional data and applies the conditional data to rules and information provided in the virus knowledge module to make determinations about the type of virus scanning to be performed. Exemplary determinations include the determination of whether a scan is appropriate, the assessment of conditions pertaining to the request for scanning, and the type of scanning and cleaning routines to be implemented.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 917

According to still another aspect of the present invention, a network diagnosis and maintenance application is provided. The network diagnosis and maintenance application gathers information about a network and maintains diagnostic data corresponding to the gathered information. Determinations about network conditions are made by applying the diagnostic data to predetermined knowledge, and request for network maintenance are requested based upon the determinations. An embodiment of the diagnosis and maintenance application includes a diagnostic data module, a maintenance requesting module, and at least one expert system with an inference engine and predetermined knowledge. The diagnostic data module provides the diagnostic data, which is accessed by the expert system and then applied to rules pertaining to its field of expertise to make determinations about the network. The determinations are then used by the maintenance requesting module as a basis for requesting network maintenance. An exemplary expert system is a virus information expert system including an inference engine and a virus knowledge module. The determinations made by the virus information expert system are communicated to the maintenance requesting module, which can use the determinations to request virus scanning and treatment by a virus detection server.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other more detailed and specific features of the present invention are more fully disclosed in the following specification, reference being had to the accompanying drawings, in which:

FIG. 1 is a schematic illustrating exemplary interconnections between a client and a virus detection server in accordance with the present invention.

FIG. 2 is a flow diagram illustrating a method of iterative virus detection in accordance with the present invention.

FIG. 3 is a block diagram illustrating an embodiment of a client in accordance with the present invention.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 918

FIG. 4A is a block diagram illustrating an embodiment of a virus detection server in accordance with the present invention.

FIG. 4B is a block diagram illustrating an embodiment of an iterative virus detection module in accordance with the present invention.

FIG. 4C is an exemplary data table illustrating indexed information corresponding to viruses.

FIG. 4D is an exemplary data table illustrating virus signature component strings used with a virus detection technique in accordance with the present invention.

FIG. 5 is an event diagram illustrating a method of virus detection using a virus detection server in accordance with the present invention.

FIG. 6A is a block diagram illustrating another embodiment of an iterative virus detection module in accordance with the present invention.

FIG. 6B is a block diagram illustrating an embodiment of a virus information expert system in accordance with the present invention.

FIG. 7 is a schematic illustrating an exemplary network communication system in accordance with the present invention.

FIG. 8A is a block diagram illustrating an embodiment of an administrative server in accordance with the present invention.

FIG. 8B is a block diagram illustrating an embodiment of a network diagnosis and treatment application in accordance with the present invention.

DETAILED DESCRIPTION OF THE INVENTION

Referring now to the schematic diagram of Fig. 1, the virus detection server and The method of virus detection of the present invention preferably operate on a system wherein a client-server relationship can be established between the virus detection server 400 and any one of a plurality of clients 300. In the system 100 illustrated in Fig. 1, the virus detection server 400 resides on a wide area network (WAN) such as the network typically referred to as the Internet or World Wide

IPR2024-00027
CrowdStrike Exhibit 1002 Page 919

Web. Various exemplary interconnections are shown between clients 300 and a virus detection server 400. In one example, the client 300a is shown coupled via line 10 to the WAN to allow Communication between the client 300a and the virus detection server 400. In this example, particularly where the WAN is the Internet, access can be provided by an internet access provider or an internet service provider and the client 300 includes a modem coupled to a telephone line to link to the service or access provider. In another example, the client 300b is part of a local area network (LAN) and communication between the client 300b and the virus detection server 400 can be facilitated by a connection 20 established through a LAN server 350. Although access can also be through an access or service provider, the connection 20 can be by other conventional means such as a leased line.

Although the preferred embodiment of the present invention contemplates that the virus detection server 400 is an Internet server, the ordinarily skilled artisan will recognize the various alternatives for establishing a client-server connection between the virus detection server 400 and a client 300, such as interconnection within a local area network of computers or over any internetwork connection of computers.

The virus detection server 400 uses a conventional operating system such as UNIX or Windows NT and implements conventional internet communication protocols such as the transmission control protocol/internet protocol (TCP/IP) suite. Additionally, the communication protocols can include various conventional application layer protocols such as the file transfer protocol (FTP) and/or the Hypertext Transfer Protocol (HTTP) for the transfer of files or other information between the client and the virus detection server 400.

After communication between the virus detection server 400 and the client 300 are conventionally established, the virus detection server 400 operates to determine whether a virus is associated with a client 300 and, where a virus is detected, to provide appropriate treatment for the type of virus that is detected.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 920

Briefly, the virus detection server 400 operates in conjunction with the client 300 to determine whether viruses reside at the client by iteratively producing virus detection objects at the server, transmitting the virus detection objects to the client, executing the virus detection objects at the client, and using the results of the execution of the virus detection objects, particularly the current virus detection object, to produce subsequent virus detection objects and ultimately to detect the presence of a virus. Additionally, a vaccine comprising one or more virus treatment objects is tailored to treat the conditions as determined by the iterative production and execution of the virus detection objects.

Referring now to Fig. 2 along with Fig. 1, a flow diagram illustrating a method of iterative virus detection in accordance with the present invention is described. Figs. 1 and 2 provide a general understanding of iterative virus detection. An embodiment of a virus detection server 400 and method for detecting viruses are described in further detail with reference to Figs. 3-5 below.

In an initial step 205, a request for a virus scan is received, typically from a source external to the virus detection server 400 such as a client to be scanned. After receipt of the request, in step 210 it is determined by the virus detection server 400 whether a scan is to be performed. Preferably, a validation of the virus scan request is performed pursuant to the determination of whether a scan is to be performed. Continuing with a typical example, the request can be provided by the client 300 in the form a request directed to the virus detection server 400, whereupon the virus detection server 400 can validate the request before proceeding with the determination of whether a virus is associated with the client 300. Preferably, request validation is made by reference to information stored at or accessible to the virus detection server 400.

The conventional protocols of the network, such as the WAN in the embodiment of Fig. 1, are used to route the request from the client 300 to the virus detection server. For example, the request can implement a uniform resource locator (URL) address corresponding to the virus detection server 400. Once

IPR2024-00027
CrowdStrike Exhibit 1002 Page 921

contact is initiated, conventional protocols, such as the HTTP and FTP referenced above, are used to transfer information between the virus detection server 400 and the client 300. Various alternatives for the establishment of a connection between the client 300 and the virus detection server 400 can be provided. The request for a virus scan can also be routed through a separate server that handles initial requests and, possibly, validates them before passing the client 300 to the virus detection server 400 (and thus the request and the validation would be received together by the virus detection server 400). Additionally, although a connection with and resultant virus detection and treatment for a single client is generally described, a single virus scan request that is received by the virus detection server 400 can cause the server 400 to detect and treat viruses on a plurality of clients 300, such as all of the clients 300b residing on a given local area network.

In addition to direct client 300 prompting, the request for a virus detection scan can be produced as a result of a triggering event. Thus, a virus detection scan can be initiated without user input, or knowledge in certain instances. As with a user prompted request, the triggering event causes a request for a virus scan to be Transmitted to the virus detection server 400, which receives the request and validates it prior to the progression of the determination of whether a virus is associated with the client 300. An agent program, resident at the client 300 or elsewhere, can be used to produce the request for a virus scan in reaction to the triggering event. In that instance, the agent program initiates the request and resultant connection between the client 300 and the server 400. The request for a virus scan, validation of the request, triggering events, and agent program are described in further detail with reference to the event diagram of Fig. 5, described below.

After it is determined 210 that a scan is to be performed, a plurality of virus detection objects are produced by the virus detection server 400 and transmitted to the client 300 until it is determined whether a virus is associated with the client 300. In step 215, an initial virus detection object is produced by the virus detection

IPR2024-00027
CrowdStrike Exhibit 1002 Page 922

server 400 and transmitted to the client 300. The virus detection objects include a program or code that the client 300 is capable of executing as well as data. Alternatively, selected virus detection objects might consist of executable code only, or data only. Conventional network transmission protocols are used to transmit the virus detection object from the server 400 to the client 300. After receipt of the virus detection object, in step 220 the virus detection object is executed by the client 300 and in step 225 the results of virus detection object execution are transmitted to the virus detection server 400 which receives the results and in step 230 produces an additional virus detection based upon the result of the execution of the first virus detection object. The extent of the results transmitted to the server 400 can vary. For example, the results can dictate the production of the additional virus detection object directly, can be raw results such as data that requires interpretation by the server 400 in order to produce the additional object, or can be intermediate to the two extremes. Additionally, the extent of the results can be structured according to the desired allocation of resources between the client 300 and the virus detection server 400. As with the initial virus detection object, the additional virus detection object is transmitted to the client 300 so that, as shown in step 235, it can be executed, and then results are transmitted to the virus detection server 400. The production of virus detection objects continues until a determination is made in step 245 as to whether additional scanning is no longer required. Preferably, the determination 245 is made by the virus detection server 400 based upon the last received result, but it is understood that the client 300 can make the determination and include the determination in the result that is transmitted to the client. If in step 245 it is determined that additional scanning is required, then the method returns to step 230 wherein an additional virus detection object is produced.

By proceeding with the determination of whether a virus is associated with the client 300 in iterative fashion, the information transmitted from the server 400 to the client 300 is tailored to include only that information necessary for the

IPR2024-00027
CrowdStrike Exhibit 1002 Page 923

detection of viruses at the client 300. Specifically, the information can be limited to detect the type of viruses that are likely to reside at the client based upon the conditions presented at the client (such as platform or file type), or could be limited to the type of virus detection request made by the client. This significantly reduces the overall amount of information necessary for the detection of viruses and, likewise, reduces the amount of information to be transmitted from the server 400 to the client 300. Additionally, the client 300 has access to state of the art virus detection information since the information used by the virus detection server 400 can be maintained without requiring client 300 action. Moreover, there is no need for the client 300 to engage in a comprehensive downloading of an entire virus detection comparison data update. Finally, computational overhead on the client side is minimized since the virus detection objects and any executable portions thereof are tailored to the conditions presented at the client 300.

If, as a result of the iterative production, transmission and execution of virus detection objects, it is determined in step 250 that a virus is not associated with the client 300, then the client 300 or other relevant party is appropriately notified in step 265 by the virus detection server 400. However, if a virus is detected, then in step 255 the virus detection server 400 produces and transmits to the client 300 a vaccine tailored for the treatment of the type of virus that was detected. Finally, in step 260, the vaccine is executed by the client. By tailoring the vaccine according to the results of the execution of the virus detection objects, the amount of information transmitted to and the computational resources consumed at the client 300 are minimized while an up to date treatment protocol is implemented. Preferably, the vaccine comprises at least one virus treatment object produced by the virus detection server 400 based upon the results of the iterative virus detection object execution. Additionally, as with the virus detection objects, a plurality of virus treatment objects can be produced to iteratively treat the client 300. Specifically, a first virus treatment object can be transmitted to the client 300 for execution to ascertain additional information, such as system information or

IPR2024-00027
CrowdStrike Exhibit 1002 Page 924

information input by the user, for the treatment of the virus. Then, a second virus treatment object is produced based upon the results of the execution of the first virus treatment object, and is transmitted to the client 300 for execution.

Referring now to the block diagram of Fig. 3, an embodiment of a client 300 used in accordance with the present invention is shown. The client 300 includes a central processing unit (CPU) 312, a memory 314, a data storage device 316 such as a hard disk, a keyboard 318, a monitor 320, a mouse 322, a network interface 324 and a communications unit 326. The CPU 312 is coupled to the memory 314, data storage device 316, the keyboard 318, the monitor 320, the mouse 322, the network interface 324 and the communications unit 326 via a system bus 310 in a conventional architecture. The CPU 312 is preferably a microprocessor such as a Pentium as provided by Intel of Santa Clara, California. Additionally, the monitor 320, keyboard 318, and mouse 322 are conventional and configured to provide and receive data as is well known. The communications unit 326 is a conventional device configured to allow communication between the client 300 and other computers. Preferably, the communications unit 326 is a conventional modem connected to a line 10 such as a conventional telephone line. The network interface 324 is also a conventional device configured to allow communication between the client 300 and other computers. Preferably, the network interface 324 is a conventional network adapter card including a transceiver that is coupled to a network transmission medium 25 such as a coaxial, twisted pair, fiber optic, or other line. As shown in Fig. 1, communication between the client 300 and the virus detection server 400 can be facilitated through the network interface 324 or, alternatively, the communications unit 326.

In conventional fashion, the CPU 312 can be directed by instructions provided in memory 314 to perform desired functions. Additionally, input can be provided by the mouse 322 and keyboard 318 and output to the monitor 320 in conventional fashion, and communication between the computer and other devices

IPR2024-00027
CrowdStrike Exhibit 1002 Page 925

can be facilitated through the communications unit 326 and/or the network interface 324 in conventional fashion.

The memory 314 is shown to include an operating system 328, a browser 330, and a virus detection shell 332. The operating system 328 is preferably a conventional one for a personal computer such as Windows95 or WindowsNT as provided by Microsoft, Inc. of Redmond, Washington. The memory 314 may also include application programs such as word processing and spreadsheet programs configured to function using the operating system 328. The browser 330 is preferably a software program for accessing and communicating with other computers. In the preferred embodiment, the browser 330 is Internet Explorer as provided by Microsoft, Inc. The client 300 also includes an engine capable of executing the programs provided in the virus detection objects produced by the virus detection server 400. Although it is understood that the engine could be provided otherwise with the client 300, in accordance with a preferred embodiment of the present invention, the engine is provided in conjunction with the browser 330, and thus the browser 330 is considered to be "enabled" for the execution of the programs provided in the virus detection objects. In the preferred embodiment, ActiveX programming tools, also as provided by Microsoft, Inc., are used in the production of the virus detection objects used in the iterative detection of viruses. ActiveX programming tools are means for producing objects, called "controls" in ActiveX terminology, which objects can include executable code that can reside at a server and can be accessed by a client. The controls can also be transmitted to the client, and, provided that the client includes the means for doing so, executed by the client. In the preferred embodiment the browser 330 is ActiveX enabled and the virus detection objects are ActiveX controls. Various alternatives can be used for the browser 330 and to provide the executable portions of the virus detection objects. For example, the browser 330 could be the Netscape Navigator as provided by Netscape, Inc. Additionally, the virus detection (and treatment)

IPR2024-00027
CrowdStrike Exhibit 1002 Page 926

objects can, for example, be applets such as Java applets. Java programming tools are provided by Sun Microsystems, Inc.

The CPU 312, at the direction of instructions received from memory 314, operates to receive virus detection and treatment objects, execute the objects, and transmit results to the virus detection server 400 for the client 300. Preferably, viruses are detected and eliminated without requiring a specific application program. Thus, provided that the client 300 can access the virus detection server 400 to make a request, such as through communications unit 326; make a request and communicate with the virus detection server 400, such as through instructions provided in conjunction with the browser 330; and execute objects produced by the virus detection server 400, such as via an enabled browser 330, viruses can be detected and treated at the client 300 using the virus detection server 400.

Since viruses can be detected and treated at the client 300 using the virus detection server 400 using the enabled browser 330 (or other means to execute the virus detection objects), viruses can be detected and treated without the virus detection shell 332.

However, although the virus detection shell 332 is optional, it can alternatively be provided in memory 314 for various purposes in accordance with the present invention. First, the shell 332, in lieu of the browser 330 can include the means necessary to execute virus detection objects. Additionally, the virus detection shell 332 can provide information used in conjunction with the detection and treatment of viruses. For example, the virus detection shell 332 can include account information or detailed virus detection configuration settings that can be provided to the virus detection server 400 without requiring iterations of object production and execution. Specifically, the virus detection shell 332 could include settings for defining the triggering event for requesting a virus scan or could specify the data storage devices, directories, files, file types, or any other information that could help identify or narrow the type of scanning that is desired by the user.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 927

Referring now to the block diagram of Fig. 4A, an embodiment of a virus detection server 400 in accordance with the present invention is shown. The virus detection server 400 includes a CPU 412, memory 414, a data storage device 416 such as a hard disk, I/O ports 418 and a network interface 420. The CPU 412 is conventional such as a Pentium Pro as provided by Intel Corporation, Santa Clara, California. The memory 414 is preferably conventional RAM but may also include conventional ROM. Additionally, the memory 414 is preferably configured to include routines for the iterative detection of viruses. The configurations are described in further detail with reference to the iterative virus detection module 450b of Fig. 4B. The data storage device 416, I/O ports 418 and network interface 420 are conventional and respectively facilitate long term data storage, input and output from the server, and communication with other computers. The CPU 412, memory 414, data storage device 416, I/O ports 418 and corresponding devices, and network interface 420 are coupled to a bus 410 in conventional fashion. The CPU 412, under the guidance and control provided by instructions received from memory 414, provides signals for sending and receiving data through the network interface 420 and transmission line 430 to facilitate communication with other devices such as a client 300. As described with reference to Fig. 1, conventional network communication protocols such as the TCP/IP protocol suite are used for such communication. Additionally, the instructions received from memory 414 allow the virus detection server 400 to produce virus detection and treatment objects and transmit the virus detection and treatment objects to another computer such as the described client 300. The instructions also facilitate the receipt of the results of the execution of virus detection and treatment objects and to use those results to produce additional objects.

Referring now to Fig. 4B, an embodiment of an iterative virus detection module ("IVDM") 450b in accordance with the present invention is shown to include a scanning module 454, a virus pattern module 456, a virus rules module

IPR2024-00027
CrowdStrike Exhibit 1002 Page 928

458, a cleaning module 460, a cleaning pattern module 462, an access managing module 464, and an access data module 466. The iterative virus detection module 450b, and its referenced modules, includes routines for receiving virus detection requests, validating requests, producing virus detection and treatment objects, receiving the results of the execution of the virus detection and treatment objects, and using the results to produce additional virus detection and treatment objects to ultimately detect viruses and treat them. The iterative virus detection module 450b is typically implemented in software, but can also be implemented in hardware or firmware.

The access managing module 464 includes routines for determining the validity of virus scanning requests which are received by the virus detection server 400. The access data module 466 is in operative communication with the access managing module 464 and includes information that is used in access validation. For example, information about users including a user identification code and an account validity indicator can be stored in the access data module 466. Thus, when a request for a virus scan is made, the IVDM 450b, in conjunction with the access managing module 464, identifies the requester and uses the data in the access data module 466 to determine whether the user identification code is listed and whether the account is valid. Various alternatives can be implemented for user identification and validation, such as one where a user initiates a request and is prompted for an identification code and a password to validate the requester's access of the virus detection server 400 for the detection and treatment of viruses. The access managing module 464 is also arranged to set up an account for a new user and therefore is configured to acquire the necessary information.

After it is determined that a valid request for scanning has been provided, virus detection objects are iteratively produced and transmitted to the requester, such as a client 300, in order to detect a virus. The scanning module 454 includes various routines that can be used in the detection of viruses. Preferably, the routines are provided for separate access so that virus detection objects can be

IPR2024-00027
CrowdStrike Exhibit 1002 Page 929

tailored to include selected scanning routines. For example, separate routines for the detection of viruses that could reside in systems using particular platforms and operating systems, in particular file types, and in particular locations are provided for separate access in the scanning module 454. Specifically, routines for the detection of viruses that typically reside on one platform are provided such that they can be accessed separate from routines for the detection of viruses that typically reside on another platform. Similarly, routines for the detection of viruses that reside in "executable" files (such as those that have the file extension *.exe*) are provided such that they can be accessed separate from routines used for the detection of macro viruses (such as those that implement the WordBasic programming language, typically reside in application data files, and include extensions such as *.doc* or *.dot*). Thus, a virus detection routine that examines a file to determine whether it includes a virus signature can be separated from a virus detection routine that use a set of rules, such as combinations of suspect instructions, to determine whether viruses are present in files. Other types of virus detection routines can be provided in the scanning module 454, such as those used in the detection of viruses in electronic mail messages.

Preferably, the routines provided in the scanning module are configured to use the particular programming tools used by the virus detection server 400 to detect viruses at the client 300. Thus, the routines are capable of being executed by the client 300. Alternatively, the routines provided in the scanning module can be converted from a first executable type to an executable type supported by the client 300.

In the preferred embodiment, various routines are provided for separate access in the scanning module 454. For example, routines corresponding to the detection of virus signatures in various types of files are provided. Information about each routine is provided in association with the routine to facilitate coherent separate access. For example, for virus signature detection routines, information such as that corresponding to platform and file type are provided in association

IPR2024-00027
CrowdStrike Exhibit 1002 Page 930

with each routine. Thus, the IVDM 450b, in conjunction with the scanning module 454, can efficiently determine which routine requires access when necessary. Typically, the decision whether to use a particular routine will depend upon the conditions at the client 300. The conditions can be predetermined or can be acquired by the virus detection server 300 through interaction corresponding to scope and risk assessment, described further below with reference to Fig. 5. Conventional indexing and sorting techniques can be used to tag the various virus scanning routines provided in the scanning module 454 and, accordingly, to facilitate separate access.

Similar to the scanning module 454, the virus pattern module 456 and virus rules module 458 respectively include virus signatures and rules that can be used in the detection of viruses. As with the routines described in connection with the scanning module 454, the patterns and rules are provided to facilitate separate access. Thus, for example, the file signatures in the virus pattern module 456 corresponding to one type of file can be separated from the virus signatures corresponding to a second file type. Similarly, the rules in the virus rules module 458 corresponding to one detection criteria can be separated from the virus rules that apply to a second detection criteria. As with the routines in the scanning module 454, conventional indexing and sorting techniques can be used to provide separate access to the patterns and rules. Of course, a plurality of signatures or rules could apply to a particular indexing field. For example, several signatures would correspond to the indexing field .exe and thus would be included, subject to other indexing limitations, in the virus signatures corresponding to such scanning conditions.

In addition to providing routines and corresponding signatures or rules that separately accessible and thus amenable to tailoring according to the assessed scope and risk presented at the client 300, the virus scanning stage is provided such that scanning can be undertaken in iterations.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 931

For example, in a preferred technique a plurality of files, such as all of those residing on a client 300 hard disk 316, are initially scanned to determine whether they include a first portion of a virus signature (or a plurality of virus signature portions). The result of the initial scan will indicate those files, if any, that include the portion. Then, a second scan is undertaken based upon the result of the first scan. Specifically, those files that include the first portion are scanned to determine whether they also include a second virus portion. Those files that include the first portion and the second portion are candidates for inclusion of the entire virus signature and thus the virus. With this technique, the amount of data used as comparison data for the initial scan is minimized since only a portion of the virus signature is required for the scan. Additionally, since the number of files to be scanned is gradually narrowed, the computational resources used in the scan, particularly at the client 300, are lowered.

An exemplary indexing of virus signatures and the preferred signature scanning technique are now described with reference to Figs. 4c-4d. Referring to Fig. 4c, an exemplary data table 475 is shown to include columns for platform, virus type, and virus identification. In the exemplary table 475, each row includes information about a particular virus. The information can be used to determine whether a scanning routine corresponding to the particular virus will implemented. Of course, various scanning routines will correspond to groups of viruses with common characteristics. The data table 475 provides an example of how various virus information is indexed. Various additional or alternative criteria for determining which scanning and treatment routines to use can be provided.

Preferably, information such as that provided in the exemplary data table 475 is provided in memory 414 for access by the IVDM 450a in the selection of virus scanning and treatment routines and, more specifically, in the production of virus detection and treatment objects. Assuming that, either by iterative object determination in the scope and risk assessment stages, or by user input, or by predetermined settings, that only files type viruses corresponding to a given

IPR2024-00027
CrowdStrike Exhibit 1002 Page 932

platform I are targeted for scanning, the target would only be scanned for viruses such as viruses A, C, and E. The scanning routines are provided in the scanning module 454 and are indexed for access. Additionally, virus signatures are provided in the virus pattern module 456 and are indexed for access. Thus, using information such as that shown in the data table columns, the appropriate scanning routines for detecting file type viruses and virus signatures corresponding to viruses A, C and E are accessed. Although a virus detection object including full signatures corresponding to these three selected viruses could be provided in accordance with the present invention, a technique is provided which allows a further reduction in the amount of information required for transmission to the client 300 in order to detect viruses.

Referring now to Fig. 4d, another exemplary data table 480 includes columns for the virus identifier and a column generally indicating segregated virus component string data. As shown, the virus signature for each identified virus is broken into portions that can be used in the detection of viruses. In one embodiment, the portions comprise one byte strings, but it is understood that the length of the string can differ. Specifically, for virus A, strings A1, A2, and A3 are shown, for virus B, strings B1, B2 and B3 are shown, and for virus C, strings C1, C2 and C3 are shown. The table 480 is provided by way of example. It is understood that many additional portions or strings could be provided for each virus. It is also understood that data for many more viruses could be provided.

An iterative, progressive virus scan can be undertaken using virus signature portion data such as that in the data table 480. There are two ways that this type of scanning can be implemented, referred to as the first and second types below.

In the first type of detection, detection is undertaken by iteratively scanning all targeted files for the relevant string portions to progressively narrow the number of viruses that could reside in the targeted files. The virus portions do not have to be adjacent strings in the signature. Once the number of viruses that is potentially

IPR2024-00027
CrowdStrike Exhibit 1002 Page 933

present is sufficiently narrowed, a full virus signature corresponding to each possibly present virus is provided so that the appropriate files can be scanned.

In the second type of detection, the scan itself is conducted using the plural virus signature portions. Preferably, the virus portions are adjacent strings in the signature, and the virus detection objects are arranged to detect presence and location within targeted files.

Specifically referring to the first type of detection, an initial virus detection object includes a routine for scanning all of the targeted files to determine whether they include the initial portions or strings A1, B1, and C1. The virus detection is produced by the segregated and indexed data and transmitted to the client 300. The client then executes the virus detection object to produce a result. The result includes an indication as to whether the portions or strings are included in the targeted files and, if so, an identification of which files included which strings. Assuming that 100 files were targeted, that no targeted file included string C1, and 5 of the targeted files included either string A1 or B1 (or both), the result of the execution of the virus detection object would indicate as such. Preferably, the identification of the 5 files remains at the client, and the result transmitted to the virus detection server 400 is that strings A1 and B1 were found (and thus no files with string C1). The virus detection server 400 then uses this result to produce a second virus detection object. In this instance, the second virus detection object would include only strings A2 and B2, since virus C was eliminated from consideration. The second virus detection object is then executed at the client. Specifically, the client scans the 5 identified files to determine whether they include the combination of A1 & A2 or B1 & B2. The results of the previous execution (to find A1, B1, etc.) could be used such that only the second strings would need to be scanned for, or, alternatively, the files could be scanned for both strings. The execution of this virus detection object could thus discount viruses A and B from consideration. Assuming that certain files were found to include both B1 & B2 but no file included both A1 and A2, a full virus signature corresponding

IPR2024-00027
CrowdStrike Exhibit 1002 Page 934

to virus B could be provided in the next virus detection object so that the targeted file or files could be scanned to determine whether the virus resides at the client 300. It is understood that using this technique the number of iterations, the number of viruses which remain possibly present, and the number of files ultimately scanned can vary. Regardless, by using a limited amount of comparison data, a screening can be undertaken such that when full virus signatures are finally transmitted to the client 300, their number and the corresponding amount of data to be transmitted are effectively minimized.

In the second type of detection, the strings comprise adjacent portions of the viruses and thus the iterative scan can also detect the virus (rather than just limit the amount of viruses that could be present followed by a full virus signature transmission). In the second type of detection, therefore, additional information about the files including strings and the location of the strings is provided. Specifically, in the second type of detection, if the initial virus detection object indicates that 5 targeted files include A1 or B1, then an identification of which files include which string is retained, as well as a marker so that a determination could be made as to whether the adjacent string A2 or B2 respectively followed string A1 or B1 in a targeted file. In this fashion, the number of viruses which are potentially present can be rapidly minimized and efforts are not duplicated since the full signature does not have to be transmitted after transmission of virus portions.

Although a preferred technique for iteratively detecting viruses and thus minimizing the amount of data to be transmitted to the client is described, other techniques can be provided. Some examples of virus detection techniques that can be modularized or subdivided to provide iterative detection are those used in the detection of unknown viruses in macros and those used in electronic mail scanning. These examples are also discussed with reference to the virus scanning stage in the event diagram of Fig. 5.

Again the routines, patterns and rules for iterative virus scanning are respectively provided for separate access in the scanning module 454, virus pattern

IPR2024-00027
CrowdStrike Exhibit 1002 Page 935

module 456 and virus rules module 458. Still referring to Fig. 4B, the IVDM 450b, in conjunction with the virus scanning module 454, the virus pattern module 456 and the virus rules module 458, iteratively produce virus detection objects based upon the results of the execution of previous virus detection objects. Since the routines in the virus scanning module 454 and the rules and patterns in their respective modules 458, 456 are provided such that they can be separated, the IVDM 450b can produce virus detection objects that include a limited amount of data that is specifically directed at the type of conditions presented at, for example, a client 300 to be scanned for viruses. For example, where it is indicated by previous iterations (or separately provided input) that only executable files that have not been compressed and reside in a particular directory are to be scanned, the IVDM 450b accesses the routines in the virus scanning module 454 that are used for the detection of viruses in executable files, accesses the virus signatures the could be presented in executable files in the virus pattern module 456 and produces a virus detection object using the information. The appropriate virus detection routines from the scanning module 454 and information from the virus pattern module 456 and/or the virus rules module 458 are used to produce the virus detection object. As described previously, the object typically implements a conventional programming language such as that provided in ActiveX controls or Java applets.

Once each virus detection object is produced, it can be transmitted from the virus detection server 400 to the client 300 using conventional network transmission protocols. The computer client 300 executes the virus detection object and transmits the results of the execution for receipt by the virus detection server 400. The IVDM 450b, in conjunction with the virus scanning module 454, uses the execution results of the virus detection object to determine whether additional virus detection objects are required, and, where no additional virus detection objects are required, whether a virus was detected. In a preferred method of operation, the iterative production of virus detection objects is divided into

IPR2024-00027
CrowdStrike Exhibit 1002 Page 936

separate stages including the scope assessment stage, the risk assessment stage, and the virus scanning stage. In the scope assessment stage, a determination as to the extent of the virus scan is made. In the risk assessment stages, determinations about the conditions presented at the requested site are used to produce objects in the virus scanning stage.

If it is determined by the IVDM 450b in conjunction with the scanning module 454, and, where necessary, the virus pattern module 456 and virus rules module 458, that a virus is present, a vaccine is tailored according to the type of treatment desired, the conditions, and the type of virus that is detected. As with the scanning routines and detection information, the routines in the cleaning module 460 and the patterns or other data in the cleaning pattern module 462 are provided for separate access so that the routines and data necessary for the treatment can be separately accessed. Again, the access can be provided by conventional indexing and sorting techniques. Virus treatment objects are produced using the IVDM 450b in conjunction with the cleaning module 460. Additionally, as with the virus detection objects, the treatment objects can be iteratively produced using the results of previous objects.

Various remedial routines are provided by the cleaning module 460. Dependent upon the conditions presented at the client 300 and the type of virus detected, the remedial routines can take immediate actions such as deleting a file, stripping out infected portions of a file, or replacing infected portions with a benign portion. In addition to these actions, which are aimed at correcting infected files, the remedial routines can provide warnings to parties that could be affected by the detected virus, and could provide virus specific indications. For example, where a virus is detected within a particular type of Java applet that was downloaded to a client 300 within a network, the network administrator can be notified so that similar entities could be blocked from entering the network.

Referring now to the event diagram of Fig. 5, a method of iterative virus detection using the virus detection server 400 is shown to include an initiation

IPR2024-00027
CrowdStrike Exhibit 1002 Page 937

stage, a scope assessment stage, a risk assessment stage, a virus scanning stage, and a remedial stage.

The initiation stage is prompted by a request 505 for a virus scan. The request can be initiated directly by a client 300 which accesses the virus detection server 400 using conventional network communication protocols. Although the triggering event 502 that prompts the request 505 is typically initiated directly by the user of the client 300, the request can alternatively be initiated by a triggering event other than user prompting or initiation. This allows for regular virus scanning without requiring user input. Additionally, a group of computers that a user might seek to manage, such as a plurality of computers residing on a LAN, can be subjected to regular virus scanning without requiring user initiation and with minimal use of network resources. Moreover, since the virus detection server 400 can be maintained separately from the stand alone client 300 or the computers 300 on a local area network, there is little or no need for virus detection data maintenance on the part of either the client 300 users or the LAN administrator. Preferably, an agent program is used to provide non-user initiated scanning requests. The agent program gathers information without necessarily requiring the presence or input of the user and initiates a request for a virus scan whenever the gathered information corresponds to a predetermined triggering event.

An exemplary triggering event is the lapsing of a predetermined amount of time since the most recent virus scan. Other triggering events include those likely to facilitate the spread of viruses. These triggering events include attempted internet access, receipt or transmission of electronic mail, or downloading files over the internet. Additionally, a plurality of events in combination, such as the first attempted internet access after a predetermined amount of time has elapsed since the most recent virus scan, can be the triggering event. It is understood that various triggering events in addition to those described above can be provided.

The agent program, upon a detection or determination that the triggering event has occurred, issues a request for a virus scan to the virus detection server

IPR2024-00027
CrowdStrike Exhibit 1002 Page 938

400. Then, the response (after validation) by the virus detection server 400 is "pushed" to the client 300. Conventional programming techniques are used for the agent program and conventional techniques are used to gather the relevant information to determine the triggering event occurrence.

Preferably, where the client 300 is not a part of a local area network such as client 300a as shown in Fig. 1, the agent program resides at the client 300a. Where the client 300 communicates with the virus detection server 400 through local area network channels such as clients 300b as shown in Fig. 1, an agent program can reside at the client 300b, or, preferably, can reside at a LAN server 350 (such as a gateway server, a non-gateway LAN server, or a proxy server) capable of monitoring network traffic relevant to the determinations corresponding to the triggering event. Preferably, the agent program is produced using routines provided in the access managing module 464. The agent program is transmitted to the selected location (e.g. client 300a, client 300b, LAN server 350, proxy server) using conventional network transmission protocols, and can be stored in the client 300 memory 314 or the storage location appropriate to the selected location. Conventional programming languages can be used for the agent program.

The request 505 for a virus scan is detected 510 by the virus detection server 400, and then it is determined 520 whether the request is valid. The routines and rules for the determination of access validity are respectively provided by the access managing module 464 and rules module 466 and are described in connection with the iterative virus detection module 450b. Briefly, the filtering module 464 identifies the request and accesses the rules in the rules module 466 to determine whether the request is valid by reference to the access rules. Existing accounts are typically filtered, but the module 464 is arranged to accept and arrange the origination of new accounts.

Once it is determined that a valid request for virus scanning has been made, the scope assessment stage commences. In the scope assessment stage, the virus detection server 400 typically determines where a scan will be performed. Various

IPR2024-00027
CrowdStrike Exhibit 1002 Page 939

information can be determined in the scope assessment stage, such as the media to be scanned, the locations (e.g. directories) to be scanned, or which group of files are to be scanned. A virus detection object to ascertain the scope of the virus scan is produced 525 by the virus detection server 400 and is then transmitted 530 to the client 300 using conventional network communication protocols. Preferably, the virus detection object prompts input by the user at the client 300 to procure information about the scan. One or more predetermined virus detection objects can be indexed and stored in memory 414 for access by the IVDM 450b. One example of a virus detection object procures information including the devices, memories, directories, and files that require scanning.

The client executes 535 the virus detection object in order to gather the relevant information. Of course, user input may be required dependent upon the requirements dictated by the virus detection object. The client 300 also transmits 540 the results for receipt by the virus detection server 400. Preferably, the virus detection object includes a portion for transmitting 540 the results to the virus detection server 400 to obviate the need for continuing user input in the exchange of objects and results. Alternatively, the user can be prompted for simple input, such as clicking a button with the mouse 322 or through keyboard 318 input, in order to cause the transmission 540.

Various alternatives can be provided for the scope assessment stage. The production of a virus detection object corresponding to the stage is not absolutely required. For example, the IVDM 450b can retain virus detection information in the access data module 466, or elsewhere, corresponding to the identification of the requester. Additionally, account information corresponding to particular types of requests can be provided in the access data module 466. The account information can correspond to all of the computers on a LAN and provide all the necessary background information about those computers so that they can be scanned without iterative scope assessment. Alternatively, the information can be provided along with the request for a virus scan, particularly where the client 300 includes a virus

IPR2024-00027
CrowdStrike Exhibit 1002 Page 940

detection shell 332. Finally, the virus detection server 400 can rely upon default settings (e.g. scan the entire client 300 hard disk) and thus proceed with a virus scan using the default settings, without inquiry.

After the necessary information is gathered, the risk assessment stage commences. In the risk assessment stage, the determined scope of scanning is analyzed to determine whether and which files or data could contain viruses. Preferably, virus detection objects are produced and transmitted to the client 300 to determine whether and what types of scanning will be required. Where a virus detection object corresponding to the scope assessment stage is provided, the results of its execution at the client 300 can be used in the production of the next virus detection object. Various information is ascertained in the risk assessment stage. Where specific targets are identified, such as specific storage media or directories, the virus detection object is produced 545 to determine whether and what type of scanning is required for the specified targets. For example, the virus detection object is produced 545 to determine the types of files residing in the specified directory (e.g. whether executable files are resident), or the type of storage medium or directory (e.g. whether the client or directory corresponds to a postal node in an electronic mail system) so that the risks presented by the specified targets are evaluated so that subsequent virus scanning can be tailored to the risks presented. As with other virus detection objects, the object includes an executable program (and possibly other data) specific to the task of making the determination.

Specifically, where, for example, a directory is initially specified as the scanning domain per scope assessment, a program for accessing the file header for each file in the specified directory, comparing the file header to predetermined data to determine whether the file is of the type that is likely to contain a virus, and maintaining an identification of those files that are found to be likely virus candidates can be provided in the virus detection object. Various alternatives for

IPR2024-00027
CrowdStrike Exhibit 1002 Page 941

the virus detection object or objects corresponding to the risk assessment stage will be evident to the ordinarily skilled artisan.

Preferably, the virus detection object is transmitted 550 to and executed 555 by the client 300 so that results corresponding to the execution can be transmitted 560 to the virus detection server 400. As with the virus detection object corresponding to the scope assessment stage, the virus detection object originally transmitted 550 to the client 300 can include the means necessary for causing the transmission 560 of the results to the virus detection server 400. Preferably, in the risk assessment stage, the transmission 560 is automatic, requiring no user input. Again, the transmissions 550, 560 between the client 300 and virus detection server 400 will implement conventional network protocols.

Although a single virus detection object is shown to correspond to the risk assessment stage, it is understood that a plurality of objects could be iteratively produced within the stage. Additionally, as with the scope assessment stage, default settings, predetermined settings or directly provided input to the virus detection server 400 can be relied upon for the risk assessment stage, obviating the requirement for virus detection objects for both the scope and risk assessment stages. In that instance, the iterative production of virus detection objects corresponds only to the virus scanning stage.

Upon completion of the risk assessment stage, the virus scanning stage commences. The virus detection server 400 iteratively produces a virus detection object or objects based upon the results of previous executions in order to scan for viruses. Thus, the virus detection objects can be specifically produced according to the desired scanning and conditions presented at the client 300. This allows effective virus scanning without requiring an exhaustive transmission of virus detection data to the client 300 to effectuate the scan. As described with reference to the iterative virus detection module 450a, particularly the scanning module 454, virus pattern module 456 and virus rules module 458, the routines, patterns and rules for the detection of viruses are provided such that they can be

IPR2024-00027
CrowdStrike Exhibit 1002 Page 942

separately accessed and thereby used to produce specific virus detection objects corresponding to the virus scanning stage. The produced virus detection objects will include executable programs according to the dictates of the information by previous virus detection object iterations or will be otherwise tailored for the conditions presented at the client 300.

Various scanning routines are implemented by the virus detection objects in the scanning stage. For example, where a signature scan of executable files is sought, the virus detection object includes a program for accessing each targeted executable file, determining whether each targeted file includes a string that corresponds to a virus signature from the group of virus signatures that could be expected to be found in the targeted files, and retaining information about the targeted files that are found to include virus signatures. Signature scanning techniques, such as a state engine, can be used determine whether targeted files include virus signatures.

One way that the amount of data to be transmitted from the virus detection server 400 to the client 300 is minimized is that only those virus signatures that could be expected according to the gathered information about the assessed scope and risk from previous stages. For example, where the scope and risk assessment indicate that only .exe files in a particular directory on a particular client drive are to be scanned, virus detection objects are tailored to effect that type of scan. Specifically, those virus signatures which could reside in .exe files and a routine for conducting a signature scan for the files at the noted location are provided in a virus detection object, the object is executed at the client 300, and the results are transmitted to the virus detection server 400 in the detection of viruses.

Although a single virus detection object is shown to correspond to the virus scanning stage, it is understood that a plurality of objects could be iteratively produced. Thus, in addition to the tailoring offered by the scope and risk assessment, the iterative production of virus detection objects can be independently

IPR2024-00027
CrowdStrike Exhibit 1002 Page 943

tailored to minimize the amount of information to be transmitted between the server and client and to minimize the use of computational resources.

Various types of scanning are implemented for efficient iterative analysis. An preferred technique which allows exhaustive signature scanning without requiring a comprehensive download of the signature data is described with reference to Fig. 4D above. Other scanning techniques are also provided for iterative virus detection. For example, techniques for the determination of whether a macro includes a virus can be modularized or divided into a plurality of functions which can be performed by separate virus detection objects. To explain, in the detection of known and unknown viruses in macros, combinations of suspect instructions can be used in the determination of whether a file includes a virus. Specifically, a macro that includes both a first suspect instruction and a second suspect instruction can be determined to include an unknown (or known) virus. Thus, in accordance with the present invention, separate virus detection objects are provided to first detect whether and which of several targeted files include a first suspect instruction, and then to determine whether those targeted files that include the first suspect instruction also include the second suspect instruction to detect a virus. To produce a first virus detection object the IVDM 450a operates with the scanning module 454 and the virus rules module 458 to obtain the routines and data required for the detection of the first suspect instruction. After transmission of the first virus detection object from the virus detection server 400 to the client 300 and execution by the client 300, identification of those files that include the first suspect instruction can be identified in the results transmitted to the virus detection server 400. Alternatively, the file identifiers can remain at the client 300 and the result could indicate which suspect instruction was identified. In either case, the virus detection server 400 can use the results to produce an additional virus detection object to determine whether the second suspect instruction is present.

Additionally, techniques for the detection of viruses in electronic mail messages can be modularized or divided to provide for iterative virus detection

IPR2024-00027
CrowdStrike Exhibit 1002 Page 944

within the virus scanning stage. Preferably, an initial virus detection object determines whether the whether any unread messages reside at the client 300 (or at a postal node used by the client 300). If the result of that determination indicates that there are unread messages, then additional virus detection objects can determine whether the unread messages include attachments, decode any found attachments, and scan the decoded attachments to determine whether the unread messages include viruses. Although a single virus detection object could be produced for each of the functions of determining whether there are unread messages, determining whether attachments are included, decoding attachments, and scanning decoded attachments, each virus detection object could alternatively perform more than one of the noted functions.

Still referring to Fig. 5, once the virus detection object is produced 565, it is transmitted 570 to the client 300 for execution 575. The results of the execution are transmitted 580 to the virus detection server 400 so that the results can be analyzed 585. As with the risk assessment stage, the virus detection object originally transmitted 570 to the client 300 preferably includes a portion for causing the transmission 580 of the results to the virus detection server 400 without requiring client 300 input by the user or otherwise. The analysis 585 includes an initial determination of whether additional virus detection objects are required for scanning, and, where no additional virus detection objects are required, a determination of whether a virus was detected.

If, as a result of the virus scanning stage, it is determined 590 that a virus was detected at the client 300, then the remedial stage commences. A vaccine in the form of a virus treatment object is tailored according to the type of treatment desired, the conditions presented at the client 300, and the type of virus that is detected. As described regarding the IVDM cleaning module 460 and cleaning pattern module 462, various remedial routines which can be accessed separately for production of the tailored virus treatment object are provided. Additionally, various remedial actions may be undertaken, including deleting a file, stripping out

IPR2024-00027
CrowdStrike Exhibit 1002 Page 945

infected portions of a file, or replacing infected portions with a benign portion. Once the virus treatment object is produced 605 it is transmitted 610 to the client 300 so that it can be executed 615 for appropriate treatment of the virus. Optionally, the results of the treatment can be transmitted 620 to the virus detection server 400 so that the results can be analyzed 625. As with the virus detection objects, conventional network transmission protocols are used to transmit the objects to and from the virus detection server 400, and, although a single iteration is shown, a plurality of virus treatment objects can be implemented to effectuate the appropriate remedy.

Referring now to the block diagram of Fig. 6A, another embodiment of a virus detection server 450b is shown. The virus detection server 450b includes a scanning module 654, virus pattern module 656, virus rules module 658, cleaning module 660, cleaning pattern module 662, access managing module 664 and access data module 666 which are similar to the modules included with the virus detection server 450a of Fig. 4b. However, in addition to these modules, the virus detection server 450b includes a virus information expert system 625. The virus information expert system 625 is in operative communication with the scanning module 654, virus rules module 658, virus pattern module 656, cleaning module 660, cleaning pattern module 662, access management module 664 and access data module 666.

The virus information expert system 625 gather information and provided conditional data, applies the conditional data to predetermined knowledge or rules, and uses the knowledge based application of the gathered information to make determinations about the propriety of virus scanning, the scope and risk presented by the scanning targets, the selection of virus detection routines, and, where viruses are detected, the selection of treatment routines.

Referring now to Fig. 6B, the virus information expert system 625 comprises an inference engine 630, a virus knowledge module 632, a user interface 634, and a conditional data module 636. The inference engine 630 is preferably a forward chaining type which determines results based upon a progression through a

IPR2024-00027
CrowdStrike Exhibit 1002 Page 946

series of rules, preferably of the condition and corresponding action type. The ordinarily skilled artisan will recognize that various alternative inference engines, such as backward chaining type and those with fuzzy logic based reasoning, can be implemented. The user interface 634 is conventional and allows users to interface with the expert system 625 to provide knowledge or other types of updates. The virus knowledge module 632 comprises the information and rules used in conjunction with the various operations of the IVDM 450b. The inference engine 630 accesses conditional data from the conditional data module 636, and applies the conditional data to the rules provided in the virus knowledge module 632 in order to make virus scanning determinations used in the operation of the IVDM 450b. Specific determinations include the determination of whether a scan is appropriate, the assessment of the conditions pertaining to the request for scanning, and the type of scanning and cleaning routines to be implemented.

The conditional data includes factual information about scanning requests, predetermined conditions pertaining to clients 300 or other entities associated with scanning requests, and, optionally, information discovered during scanning operations. Additionally global information pertaining to all of the virus scanning performed by the IVDM 450b can be provided in the conditional data module 636 and used in virus scanning determinations. The information pertaining to a particular account or requester can be provided by network communication with the relevant party, such as a client 300 user or a network administrator. Additionally, information can be gathered by agent programs and provided to the conditional data module 636 without requiring user input.

In the preferred embodiment, the condition and action based rules in the virus knowledge module 632 are provided in an if ... then format and are classified to facilitate knowledge updates. Preferred rule classes include those pertaining to scanning interval, condition diagnosis, and remedies. The scanning interval rules are used in the determination of whether scanning is to be undertaken, the diagnostic rules are used in the determination of whether viruses may be present

IPR2024-00027
CrowdStrike Exhibit 1002 Page 947

and which scanning routines are appropriate dependent upon the conditions present, and the remedial rules are used to determine the appropriate response to the detection (or lack of detection) of a virus.

The scanning interval rules preferably include temporal and circumstantial conditions and corresponding actions. Some exemplary scanning interval rules are provided to illustrate the various conditions that could be determined and reacted to in accordance with the present invention: "if scanning has not been undertaken in timing interval a, then scan for viruses", "if targeted client has attempted internet access n times, then divide timing interval by 2", "if targeted client has attempted internet access o times, then scan for viruses", "if client has received p electronic mail messages with attachments, then scan for viruses", "if client has downloaded files from internet, then scan for viruses", "if a virus was detected at entity in communication with client, then scan for viruses", "if client has only communicated within local area network within interval b, then multiply timing interval by 2", "if previous virus scan was within c, then do not scan for viruses", and "if scanning has not been undertaken since time d, then scan for viruses."

The exemplary rules provide for virus scans at regular intervals and can adjust the interval, in either direction, dependent upon various conditions. Additionally, floor and ceiling intervals (c,d) are provided such that scanning is mandatory within certain intervals and such that unduly repetitive scanning is not undertaken. The ordinarily skilled artisan will recognize the various conditions and actions that can be applied in the determination of whether a virus scan is to be undertaken.

The conditional data module 636 provides the information applied to the rules in the knowledge module 632 by the inference engine 630. In the preferred embodiment, the conditions are initially provided in the conditional data module 636 and are updated by communication from a client, client agent, or other entity responsible for providing the information to the virus detection server 400. The conditions are periodically applied to the rules, and, under the appropriate

IPR2024-00027
CrowdStrike Exhibit 1002 Page 948

conditions, a "request" for a virus scan is produced by the virus information expert system 625. As with the other IVDM 450a embodiment, the request can be received and validated using the access management module 664 and access data module 666. Various alternatives can be provided for the maintenance of conditional data and for the receipt and validation of requests. Specifically, the conditional data can be maintained and provided from resources other than the IVDM 450b. For example, a client 300, an agent program, or a network administrator could gather the appropriate information and transmit the information to the virus detection server 400. Further, information pertaining to a network could be provided in the form of a virus scanning request and the expert system 625 used to determine which clients 300 or other entities require scanning.

The diagnostic rules provided in the virus knowledge module 632 are used to make determinations analogous to those provided in the scope and risk assessment stages described with reference to the event diagram of Fig. 5. The diagnostic rules are applied to conditional data to determine, for a client, a group of clients, or other scannable entities corresponding to a virus scanning request, actions such as where scanning is to be performed and what type of scanning is to be performed. As with the scanning interval rules, the inference engine 630 applies the information in the conditional data module 636 to the rules in the virus knowledge module 632. The conclusions dictate the application of appropriate virus scanning routines, provided in the scanning module 654, using appropriate patterns and virus scanning rules.

The diagnostic rules preferably include circumstantial conditions and corresponding actions. The following exemplary diagnostic rules illustrate the various basic conditions that could be determined and reacted to in accordance with the present invention:

"if the scanning scope is indicated by the request for virus scanning, scan according to the request";

IPR2024-00027
CrowdStrike Exhibit 1002 Page 949

"if the scanning scope if not indicated by the request, determine whether the client includes writable media";

"if client does not include writable media, then do not perform virus scan";

"if a related scan detected Virus X, then scan client for Virus X";

"if the client includes a hard disk, then scan according to the conditions specific to the hard disk":

"if *.exe* files are present, then perform file virus signature scan",

"if *.com* files are present, then perform file virus signature scan",

"if application data files are present, then scan for macro viruses",

"if the client is a postal node, then scan all unread messages for attachments",

"if the client has received electronic mail, scan unencoded portions thereof", and

"if client includes a Java applet execution engine, then scan for hostile Java applets";

"if the client includes other writable media, the determine the conditions specific to the writable media and scan the writable media."

The exemplary rules determine the basic scope of the scan based upon the conditions prevailing at the client. The scan can also be tailored to the conditions that prompted the request for a virus scan. For example, where the scan is triggered by voluminous receipt of electronic mail, only electronic mail messages might be scanned in lieu of the entire hard disk, particularly where the hard disk has been recently scanned.

Upon determination of a valid scan request and an assessment of the conditions presented at the client 300 by the virus information expert system 625, the identified targets are scanned for viruses. Preferably, the scanning is undertaken by the IVDM 450b using the scanning module 654, in communication with the virus pattern module 656 and virus rules module 658 using iteratively produced virus detection objects using the techniques described with reference to

IPR2024-00027
CrowdStrike Exhibit 1002 Page 950

the IVDM 450a of Fig. 4B above. For example, where the virus information expert system 625 applies the conditions presented at a client 300 to the rules in the virus knowledge module 632 and it is determined that an entire hard disk is to be scanned for file type (e.g. exe, .com) viruses, then the IVDM 450b produces virus detection objects that are used to iteratively determine whether the client 300 hard disk includes any file type viruses. Preferably, the virus signature portion techniques described with reference to Fig. 4D are used to determine whether any file type viruses reside at the client 300. Although iterative detection and corresponding techniques are preferred, it is understood that the virus information expert system 625 can prompt any type of scanning. For example, the virus information expert system 625 can prompt the IVDM 450b to download comprehensive signature scanning and corresponding detection signatures to the client 300, if desired. The indexing provided in the scanning, virus pattern and virus rules modules is used to ensure that the downloaded routines correspond to the desired type of detection.

If a virus is detected as a result of the virus scan, then the remedial stage commences and the routines provided by the cleaning module 660 are implemented, along with the cleaning pattern module 662, if necessary, to provide a remedy for the detected conditions. Preferably, virus treatment objects are produced to provide the remedy as described with reference to the IVDM 450a above. Optionally, the virus information expert system 625 provides knowledge based determinations regarding the types of remedial routines to be implemented. For example, the virus information expert system 625 determines whether files will be corrected to maintain their integrity, deleted entirely, or stripped of any viral portions. Additionally, determinations are made regarding the notification provided to the client 300 user, related users, network administrator, or other parties. Furthermore, other remedial actions, such as requesting a transfer prevention for certain classes of files corresponding to the type of virus detected can be implemented. Specifically, the IVDM 450b, at the request of the expert

IPR2024-00027
CrowdStrike Exhibit 1002 Page 951

system 625 can send an email to a network administrator indicating that a certain type of file included a virus and was transmitted into the network, or can communicate directly with a network computer, such as a server including a proxy server, to provide reconfiguration instructions to block the entry of the types of files in which a virus was detected.

Referring now to the schematic diagram of Fig. 7, an exemplary network communication system 700 includes a local area network (LAN) with clients 300c, a gateway server 710 and an administrative server 750. The gateway server 710 operates an entrance to another network, here the WAN, and the administrative server 725 is a server, within the LAN, that performs administrative functions. As with Fig. 1, the virus detection server 400 resides on the WAN, and a client server relationship can be established between the virus detection server and the clients 300c or, alternatively, between the virus detection server 400 and the gateway server 710 or administrative server 725 (with the gateway server 710 or administrative server 725 being the client). The connections can be established using conventional media and conventional network transmission protocols.

Referring now to the block diagram of Fig. 8A, an embodiment of an administrative server 725 in accord with the present invention comprises a CPU 802, memory 804, data storage device 806, I/O ports 808 and a network link 810. The CPU 802 is a conventional processor, the memory 804 is conventional addressable memory (such as RAM), and the I/O ports 808 and network link 810 are conventional. The CPU 802, memory 804, I/O ports 808 (and corresponding devices), and network interface 810 are coupled to a bus 814 in conventional fashion. The CPU 802 performs functions under the guidance and control provided by instructions received from memory 804, the functions including communications through network media 812 using the network interface 810. The memory 804 also includes a proxy server 820. The proxy server 804 acts on behalf of other servers or clients in informational exchanges. Although various proxy server 804 types can be provided, a preferred proxy server 804 gathers Internet

IPR2024-00027
CrowdStrike Exhibit 1002 Page 952

requests from network computers such as clients 300c, communicates the requests to Internet servers, and receives responses from the Internet servers for communication with the requester (such as the clients 300c).

Referring now to the block diagram of Fig. 8B, the memory 804 is shown to further include an embodiment of a network diagnosis and maintenance application 825 in accordance with the present invention. The network diagnosis and maintenance application 825 can reside separate from the proxy server 804 or can be bundled with the proxy server 804. The network diagnosis and maintenance application 825 comprises a virus information expert system 830, a diagnostic data module 850 and a maintenance requesting module 875. The application 825 also includes optional additional expert systems 840, 845, such as a network response expert system and an application software expert system. The application 825 includes routines for gathering information pertaining to the network, producing and maintaining diagnostic data sing the gathered information, making determinations concerning network conditions by applying the diagnostic data to predetermined knowledge, and generating requests for network maintenance based upon the determinations. The network diagnosis and maintenance application 825 is preferably computer software, but can also be hardware or firmware.

Generally, the application 825 operates as follows. The diagnostic data module 850 includes routines for gathering information pertaining to the network and producing diagnostic data from the gathered information. The diagnostic data module 850 is in communication with the expert systems 830, 840, 845 and thus facilitates their access of the diagnostic data. Preferably, each expert system 830, 840, 845 includes an inference engine, a knowledge module, and an interface. The interface is a conventional type interface, such as a menu driven one, which allows users to communicate with the system. The knowledge module provides a knowledge base, typically in the form of information and rules which are specific to the task to be performed. The inference engine is a conventional type such as a forward chaining type. The inference engine accesses conditional data in the

IPR2024-00027
CrowdStrike Exhibit 1002 Page 953

diagnostic data module 850 and applies the conditional data to the information and rules in the knowledge module to make determinations about conditions on the network. The expert systems make specific diagnostic determinations pertaining to their field of expertise. The expert systems are also in communication with the maintenance requesting module 875, and provide their diagnostic determinations to the module 875, which produces requests for maintenance. Typically, requests are made to resources outside of the local area network, such as a virus detection server 400, to provide the maintenance.

An exemplary expert system is the virus information expert system 830. This system 830 analyzes network activity to conditional determinations including when virus scanning is to be performed, where virus scanning is to be performed, the general type of virus scanning to be performed, and the preferred remedies where viruses are detected. These determinations are provided to the maintenance requesting module 875 which communicates a request for virus scanning to the virus detection server 400 and provides the virus detection server with the conditional determinations. Other expert systems 840, 845 could be provided in accordance with the present invention. For example, a network response expert system 840 could monitor network activity to ensure response time adequacy, could make determinations as to whether outside resources should be sought to solve any inadequacies, such as by lowering the burden on local resources. Additionally, an application software expert system 845 could determine the conditions for providing software updates. Again, the determination that an update is required could prompt a request for the update from remote resources. Although exemplary additional expert systems, such as a network response expert system 840 and an application software expert system 845 are disclosed, the artisan will recognize various alternative expert systems.

Referring again to the virus information expert system 830, the virus knowledge module 834 has various rule classes including those for scanning interval and condition diagnosis. The scanning interval rules are used to determine

IPR2024-00027
CrowdStrike Exhibit 1002 Page 954

whether scanning is to be undertaken, and include temporal and circumstantial conditions and corresponding actions. The diagnostic rules also include circumstantial conditions and corresponding actions and are used to make determinations analogous to those provided in the scope (e.g. which media, directories, files are to be scanned) and risk assessment (e.g. which files or data in the defined scope might contain viruses) stages of virus treatment. Exemplary scanning interval and condition diagnosis rule sets are described with reference to the IVDM 450b resident expert system 625. Additionally, as with that system, rules pertaining to the selection of remedies upon the detection of a virus can be provided.

The diagnostic data module 850 is in communication with the proxy server 820 and is coupled to the local network transmission medium 812 to gather information about the network. Preferably, the information includes which clients 300c have attempted internet access, what types of operations and protocols were used in communications between each client 300c and other entities including those outside the network. This information is readily ascertainable by analysis of the data packets traversing the network medium or by communication directed from, for example, the proxy server 820 to the diagnostic data module 850. For example, an HTTP proxy server arranged to handle all HTTP requests by the clients 300c could merely identify the client and corresponding request to the diagnostic data module, and an FTP proxy server arranged to handle file transfers could indicate which clients 300c obtained file downloads using the relevant protocol as well as an identity of the source of the transferred file. In addition to the gathered information, background information about each client 300c, such as that concerning its CPU and storage media (e.g. hard disk) is provided in the diagnostic data module 850. Furthermore, related information such as the timing of the most recent virus scan for each client 300c is provided. The diagnostic data which is applied by the expert system to the knowledge base comprises all of this information, which is conventionally stored in the diagnostic data module 850. Of

IPR2024-00027
CrowdStrike Exhibit 1002 Page 955

course, other information is provided dependent upon the types of additional expert systems 840, 845 which are provided.

The expert systems 830, 840, 845 access the diagnostic data and make determinations concerning network conditions by applying the diagnostic data to the predetermined knowledge. Specifically, with the virus information expert system 830, the inference engine 832 applies the diagnostic data such as that relating to last scan, electronic mail receipt, file downloading, attempted internet access to the scanning interval rules provided in the virus knowledge module 834 in order to determine when a virus scan should be requested and applies the diagnostic data such as that relating to storage media and types of files downloaded in order to determine the extent of any requested virus scan.

The expert systems 830, 840, 845 are also in communication with the maintenance requesting module 875. Thus, once a determination is made that network maintenance is required, the systems 830, 840, 845 prompt the module 875 to generate a corresponding request. Specifically, in conjunction with the virus information expert system 830, the maintenance requesting module 875 directs a request for virus scanning to the virus detection server 400. In a simple case, the request pertains to a single client 300c for a comprehensive scan. However, all or selected clients 300c, storage devices, or other entities residing on the network could be scanned. To make a particular request for scanning, including which media, directories or files are to be scanned, what types of files are to be scanned, and what remedies are sought, the maintenance requesting module 875 includes an virus detection and treatment interface 880 for providing customized communication with the virus detection server 400.

The initial request, determination of request validity, and subsequent scanning stage are described with reference to the virus detection server 400. Although iterative virus detection techniques are preferably arranged and performed by a WAN resident virus detection server 400, it is understood that any resources using any scanning and treatment protocols can be implemented at the

IPR2024-00027
CrowdStrike Exhibit 1002 Page 956

request of the network diagnosis and treatment application 825. For example, the request for virus scanning by the application 825 can initiate the downloading of comprehensive virus signatures and/or scanning routines, or the performance of a scan upon the targeted client 300c by a LAN resident server using conventional signature scanning techniques.

Once the virus scanning has been performed, the determination as to whether a virus was detected, and the remedial result can be communicated to the network diagnosis and maintenance application 825, preferably using the treatment interface 880. As with the remedial stage described above, the remedy can include file deleting, file stripping, file amending, and various types of party notification. The results can be used to update the diagnostic data module 850 so that future scans can account for the determinations of previous scans. Additionally, various network elements, including the proxy server 820, can be notified or reconfigured as a result of the scan. Specifically, if a particular file corresponding to a particular transfer protocol was found to contain a virus, the appropriate proxy server can be reconfigured to block passage of that file type in the future. Additionally, using the source identification provided in the diagnostic data module 850, future file transfers from the source of the infected file could be blocked, the source could be notified, or the clients 300c on the network could be notified about the source of the infected file.

Although the present invention has been described with reference to certain preferred embodiments, those skilled in the art will recognize that various modifications may be provided. For example, although separate modules for access managing, scanning and cleaning are shown and described, it is understood that the various processes may be integrated into common modules or subdivided into additional modules which perform equivalent functions for receiving virus scanning requests, validating them, diagnosing and detecting viruses, and treating viruses. Additionally, although certain iterations in the detection and treatment of viruses are described, it is understood that the correspondence between each

IPR2024-00027
CrowdStrike Exhibit 1002 Page 957

iteration and the function or functions performed in each iteration can vary. These and other variations upon and modifications to the described embodiments are provided for by the present invention which is limited only by the following claims.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 958

CLAIMS

What is claimed is:

1. For use in a system wherein a client and a server can be connected to communicate with each other, a method for detecting viruses, the method comprising:

transmitting a first virus detection object from the server to the client;

receiving a result based upon the execution of the first virus detection object;

producing a second virus detection object based upon a result of the execution of the first virus detection object; transmitting the second virus detection object from the server to the client;

receiving a result based upon the execution of the second virus detection object; and

determining whether a virus is present at the client based upon the result of the execution of the second virus detection object.

2. The method of claim 1, further comprising:

if it is determined that a virus is present at the client, tailoring a remedy for the virus based upon at least one of the result of the execution of the first virus detection object and the result of the execution of the second virus detection object.

3. The method of claim 2, wherein the step of tailoring a remedy comprises:

producing a virus treatment object and transmitting the virus treatment object from the server to the client; and

executing the virus treatment object at the client.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 959

4. The method of claim 1, further comprising:

determining whether a virus scan is to be performed prior to the step of transmitting a first virus detection object from the server to the client.

5. The method of claim 4, wherein the step of determining whether a virus scan is to be performed comprises detecting a request for a virus scan and determining the validity of the virus scan request.

6. The method of claim 5, wherein the request is prompted by a triggering event.

7. The method of claim 6, wherein the triggering event is the first attempted internet access by the client after a predetermined time interval from a previous virus scan.

8. The method of claim 1, wherein the first virus detection object and the second virus detection object correspond to a virus scanning stage.

9. The method of claim 8, wherein the first virus detection object is used to determine whether a first virus signature portion is present in targeted files and the second virus detection object is used to determine whether a second virus signature portion is present in those targeted files that include the first virus string portion.

10. For use in a system wherein a client and a server can be connected to communicate with each other, an apparatus for detecting viruses, the apparatus comprising:

means for transmitting a first virus detection object from the server to the client;

IPR2024-00027
CrowdStrike Exhibit 1002 Page 960

means for receiving a result based upon the execution of the first virus detection object;

means for producing a second virus detection object based upon a result of the execution of the first virus detection object;

means for transmitting the second virus detection object from the server to the client;

means for receiving a result based upon the execution of the second virus detection object; and

means for determining whether a virus is present at the client based upon the result of the execution of the second virus detection object.

11. The apparatus of claim 10, further comprising:

means for tailoring a remedy for a detected virus based upon at least one of the result of the execution of the first virus detection object and the result of the execution of the second virus detection object.

12. The apparatus of claim 10, further comprising:

means for detecting a request for a virus scan; and

means for determining the validity of the virus scan request.

13. The apparatus of claim 12, wherein the request is prompted by a triggering event.

14. The apparatus of claim 13, wherein the triggering event is the first attempted internet access by the client after a predetermined time interval from a previous virus scan.

15. The apparatus of claim 10, wherein the first virus detection object and the second virus detection object correspond to a virus scanning stage.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 961

16. The apparatus of claim 15, wherein the first virus detection object is used to determine whether a first virus signature portion is present in targeted files and the second virus detection object is used to determine whether a second virus signature portion is present in those targeted files that include the first virus string portion.

17. A virus detection server for detecting viruses, the virus detection server comprising:

a processor; and

a memory, in communication with the processor, the memory providing instructions which, when executed by the processor, cause the processor to transmit a first virus detection object from the virus detection server to a client; receive a result based upon the execution of the first virus detection object; produce a second virus detection object based upon a result of the execution of the first virus detection object; transmit the second virus detection object from the server to the client; receive a result based upon the execution of the second virus detection object; and determine whether a virus is present at the client based upon the result of the execution of the second virus detection object.

18. The virus detection server of claim 17, wherein the first virus detection object is used to determine whether a first virus signature portion is present in targeted files and the second virus detection object is used to determine whether a second virus signature portion is present in those targeted files that include the first virus string portion.

19. A network diagnosis and maintenance apparatus for diagnosing and maintaining a network, the apparatus comprising:

IPR2024-00027
CrowdStrike Exhibit 1002 Page 962

a diagnostic data module, for providing diagnostic data pertaining to the network;

an expert system, in communication with the diagnostic data module, for providing knowledge for the diagnosis and maintenance of the network, for applying the diagnostic data to the knowledge, and for making determinations based upon the application of the diagnostic data to the knowledge; and

a maintenance requesting module, in communication with the expert system, for requesting network maintenance based upon the determinations made by the expert system.

20. The network diagnosis and maintenance apparatus of claim 19 wherein the expert system is a virus information expert system which comprises:

a virus knowledge module, for providing virus scanning rules, the virus scanning rules including interval rules for determining whether a virus scan should be requested and diagnostic rules for determining the scope and type of virus scanning; and

an inference engine, in communication with the virus knowledge module, for accessing the diagnostic data, for applying the diagnostic data to the virus scanning rules, and for making virus scanning determinations based upon the application of the diagnostic data to the rules.

21. The network diagnosis and maintenance apparatus of claim 20 wherein the apparatus resides within a local area network, and the maintenance requesting module directs a request for virus scanning to a virus detection server that resides outside of the local area network.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 963

22. For use with a network diagnosis and maintenance apparatus for diagnosing and maintaining a network, a method for diagnosing and maintaining networks, the method comprising:

providing diagnostic data pertaining to the network;

providing knowledge for the diagnosis and maintenance of the network;

applying the diagnostic data to the knowledge;

making determinations based upon the application of the diagnostic data to the knowledge; and

requesting network maintenance based upon the determinations.

23. The method of claim 22, further comprising:

providing virus scanning rules, the virus scanning rules including interval rules for determining whether a virus scan should be requested and diagnostic rules for determining the scope of virus scanning;

applying the diagnostic data to the virus scanning rules; and

making virus scanning determinations based upon the application of the diagnostic data to the rules.

24. The method of claim 23, wherein the request for virus scanning is directed to a virus detection server based upon the step of making virus scanning determinations.

25. A virus detection server capable of communicating with a client over a network, the virus detection server comprising:

an access managing module, for receiving requests for virus scanning, the request corresponding to a client to be scanned for viruses;

a virus scanning module, for providing routines used in the detection of viruses; and

IPR2024-00027
CrowdStrike Exhibit 1002 Page 964

a virus information expert system, in communication with the access managing module and the virus scanning module, for providing conditional data pertaining to clients to be scanned for viruses; for providing virus scanning rules including interval rules for determining whether a virus scan should be requested and diagnostic rules for determining the scope and type of virus scanning; for applying the conditional data to the virus scanning rules; and for making virus scanning determinations based upon the application of the conditional data to the rules.

26. The virus detection server of claim 25, further comprising:

a virus cleaning module, in communication with the virus information expert system, for providing routines used in the treatment of viruses.

27. The virus detection server of claim 26, further comprising:

means for selecting routines for the detection and treatment of viruses based upon the virus scanning determinations made by the virus information expert system and transmitting the selected routines to the client.

28. For use with a virus detection server capable of communicating with a client over a network, a method for detecting viruses, the method comprising:

receiving requests for virus scanning, the request corresponding to a client to be scanned for viruses;

providing a plurality routines used in the detection of viruses;

providing conditional data pertaining to clients to be scanned for viruses;

IPR2024-00027
CrowdStrike Exhibit 1002 Page 965

providing virus scanning rules including interval rules for determining whether a virus scan should be requested and diagnostic rules for determining the scope and type of virus scanning;

applying the conditional data to the virus scanning rules; and

making virus scanning determinations based upon the application of the diagnostic data to the rules.

29. The method of claim 28, further comprising providing a plurality of routines used in the treatment of viruses.

30. The method of claim 29, further comprising:

selecting routines for the detection and treatment of viruses based upon he step of making virus scanning determinations; and

transmitting the selected routines to the client.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 966

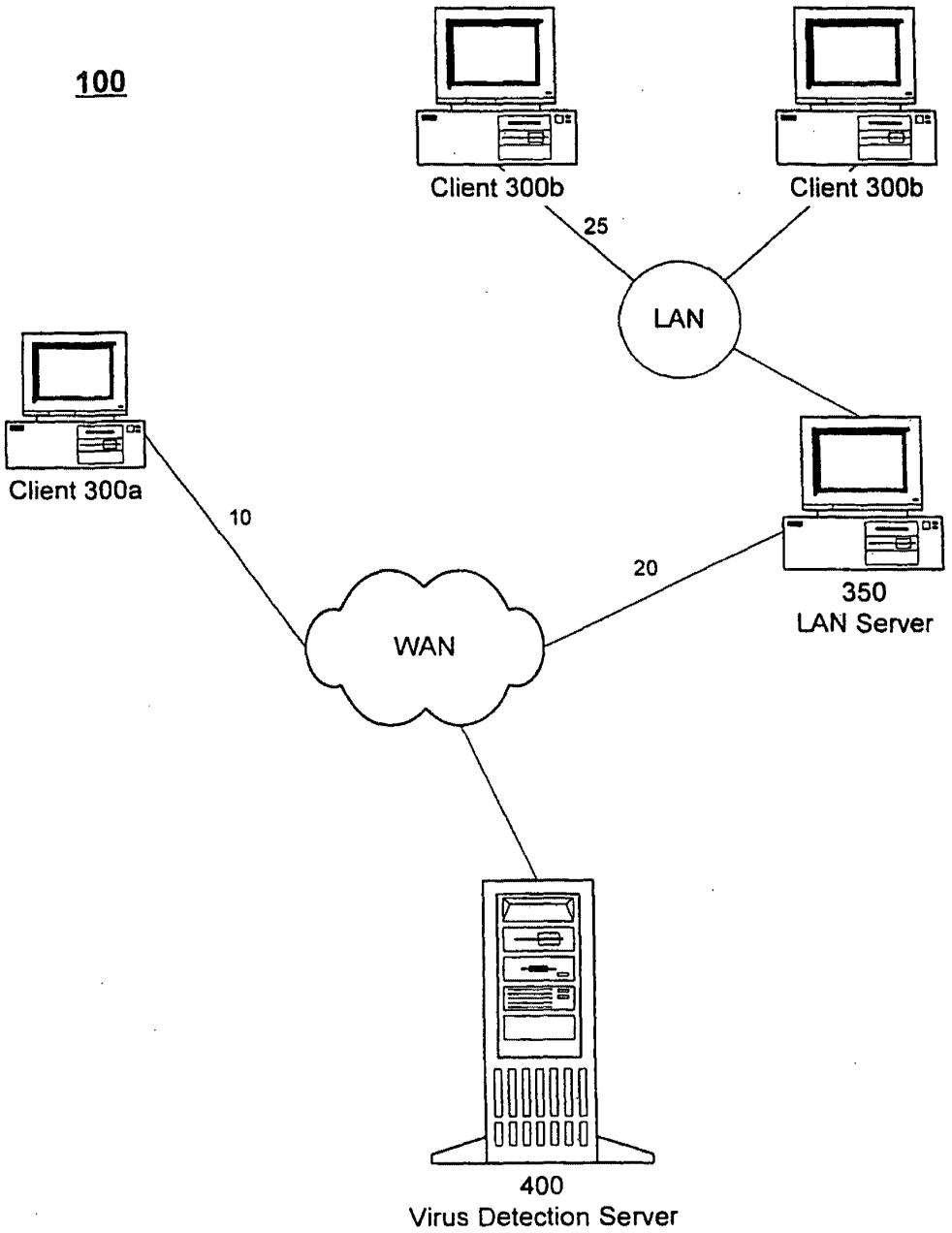

100
Client 300b
Client 300b
25
LAN
Client 300a
10
20
350
LAN Server
WAN
400
Virus Detection Server

Fig. 1

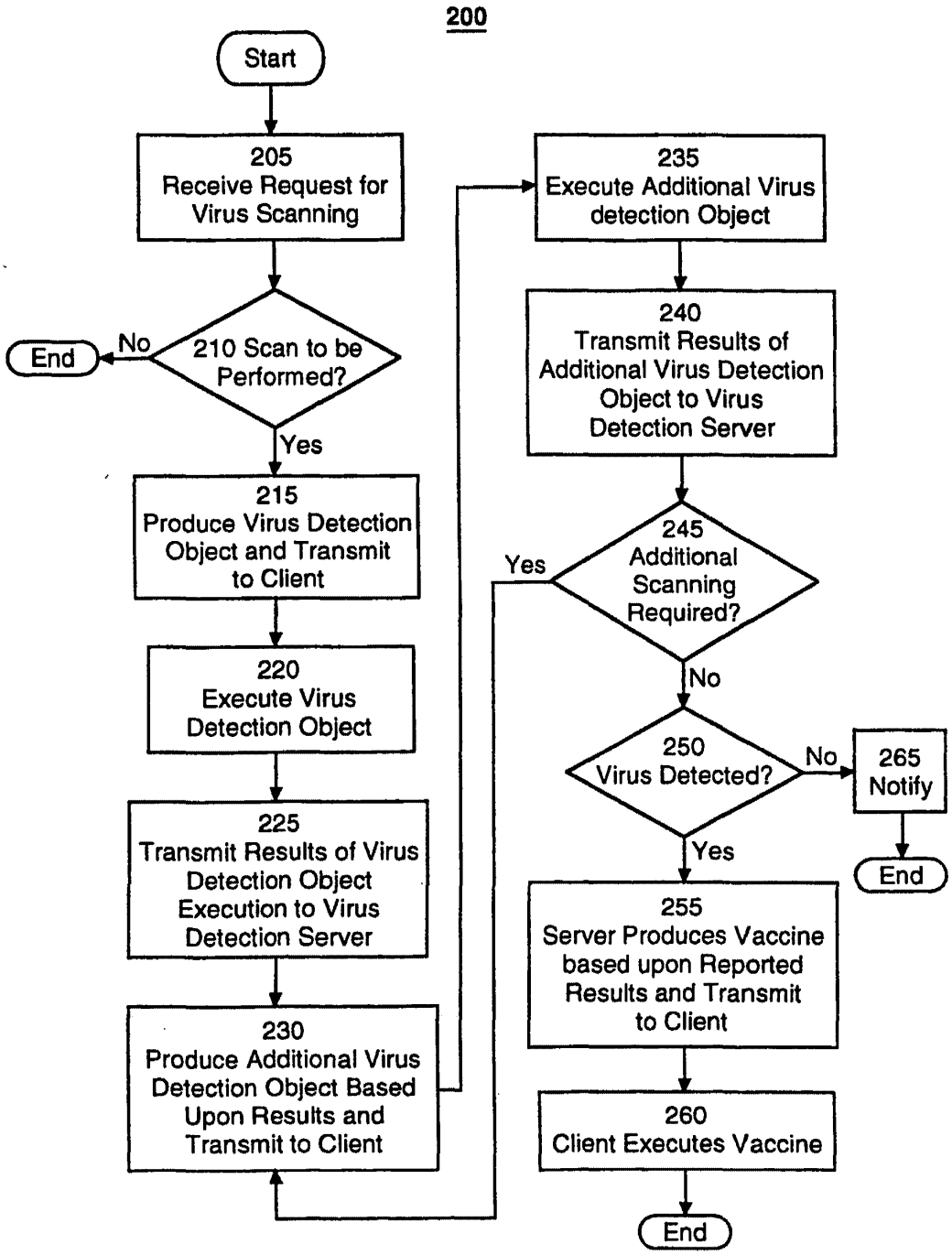

200
Start
205
Receive Request for Virus Scanning
210 Scan to be Performed?
No
End
Yes
215
Produce Virus Detection Object and Transmit to Client
220
Execute Virus Detection Object
225
Transmit Results of Virus Detection Object Execution to Virus Detection Server
230
Produce Additional Virus Detection Object Based Upon Results and Transmit to Client
235
Execute Additional Virus detection Object
240
Transmit Results of Additional Virus Detection Object to Virus Detection Server
245
Additional Scanning Required?
Yes
No
250
Virus Detected?
No
265
Notify
End
Yes
255
Server Produces Vaccine based upon Reported Results and Transmit to Client
260
Client Executes Vaccine
End

Fig. 2

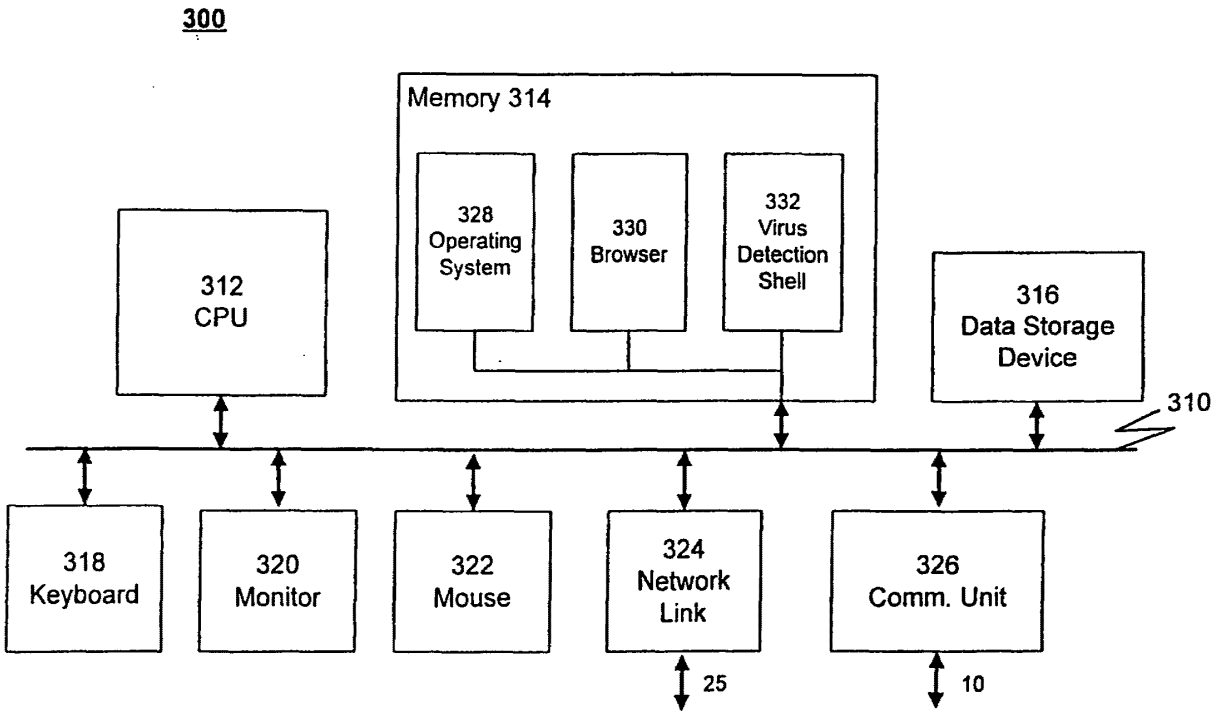

300
Memory 314
328 Operating System
330 Browser
332 Virus Detection Shell
312 CPU
316 Data Storage Device
310
318 Keyboard
320 Monitor
322 Mouse
324 Network Link
326 Comm. Unit
25
10

Fig. 3

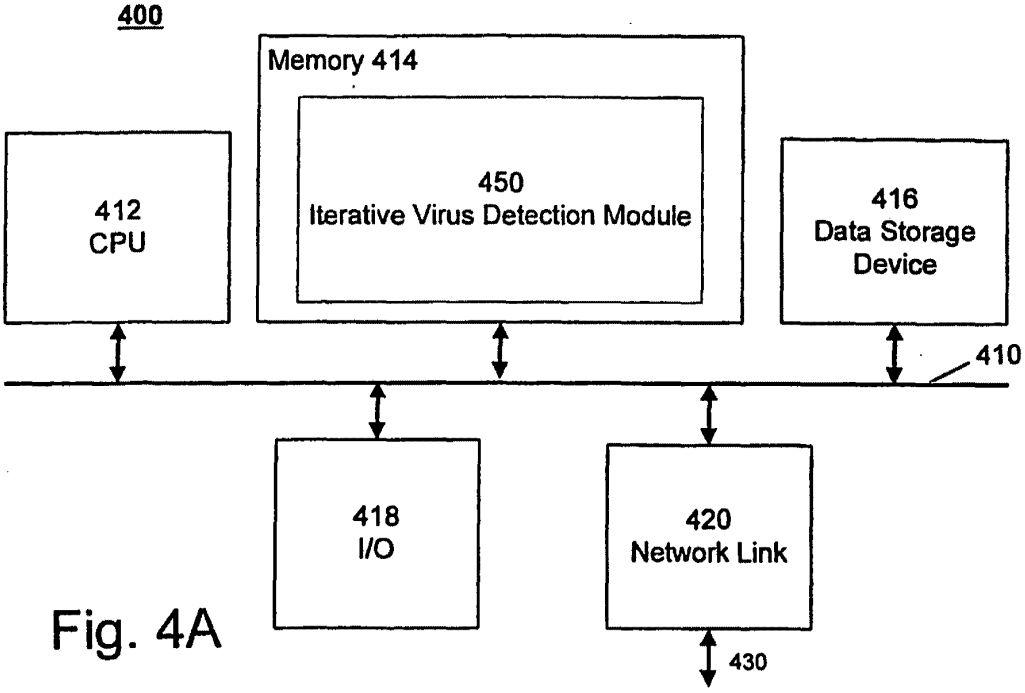

400
Memory 414
450
Iterative Virus Detection Module
412
CPU
416
Data Storage
Device
410
418
I/O
420
Network Link
430

Fig. 4A

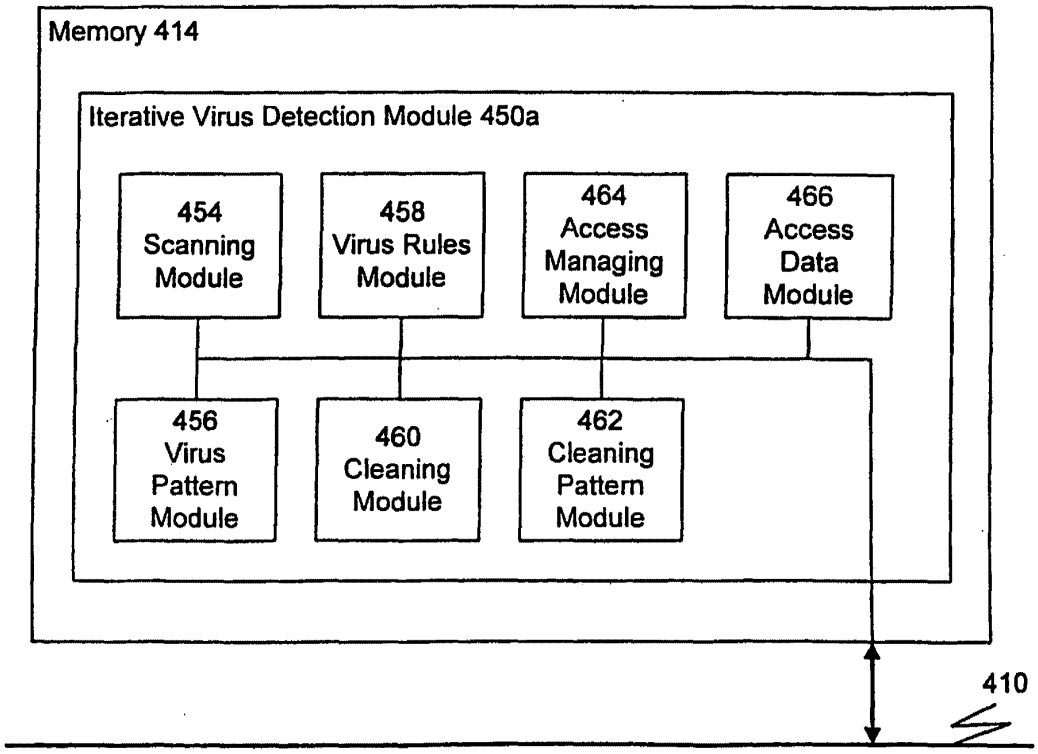

Memory 414
Iterative Virus Detection Module 450a
454
Scanning
Module
458
Virus Rules
Module
464
Access
Managing
Module
466
Access
Data
Module
456
Virus
Pattern
Module
460
Cleaning
Module
462
Cleaning
Pattern
Module
410

Fig. 4B

475

Platform	Virus Type	Virus Identifier
I	file type (e.g. *.exe, .com*)	Virus A
I	application data file (e.g..*doc*)	Virus B
I	file type	Virus C
I	applet (e.g. Java)	Virus D
I	file type	Virus E
II	file type	Virus F

Fig. 4c

480

Virus Identifier	Segregated Virus Component String Data			
Virus A	String A1	String A2	String A3	...
Virus C	String B1	String B2	String B3	...
Virus E	String C1	String C2	String C3	...
...	...	...	...	...

Fig. 4d

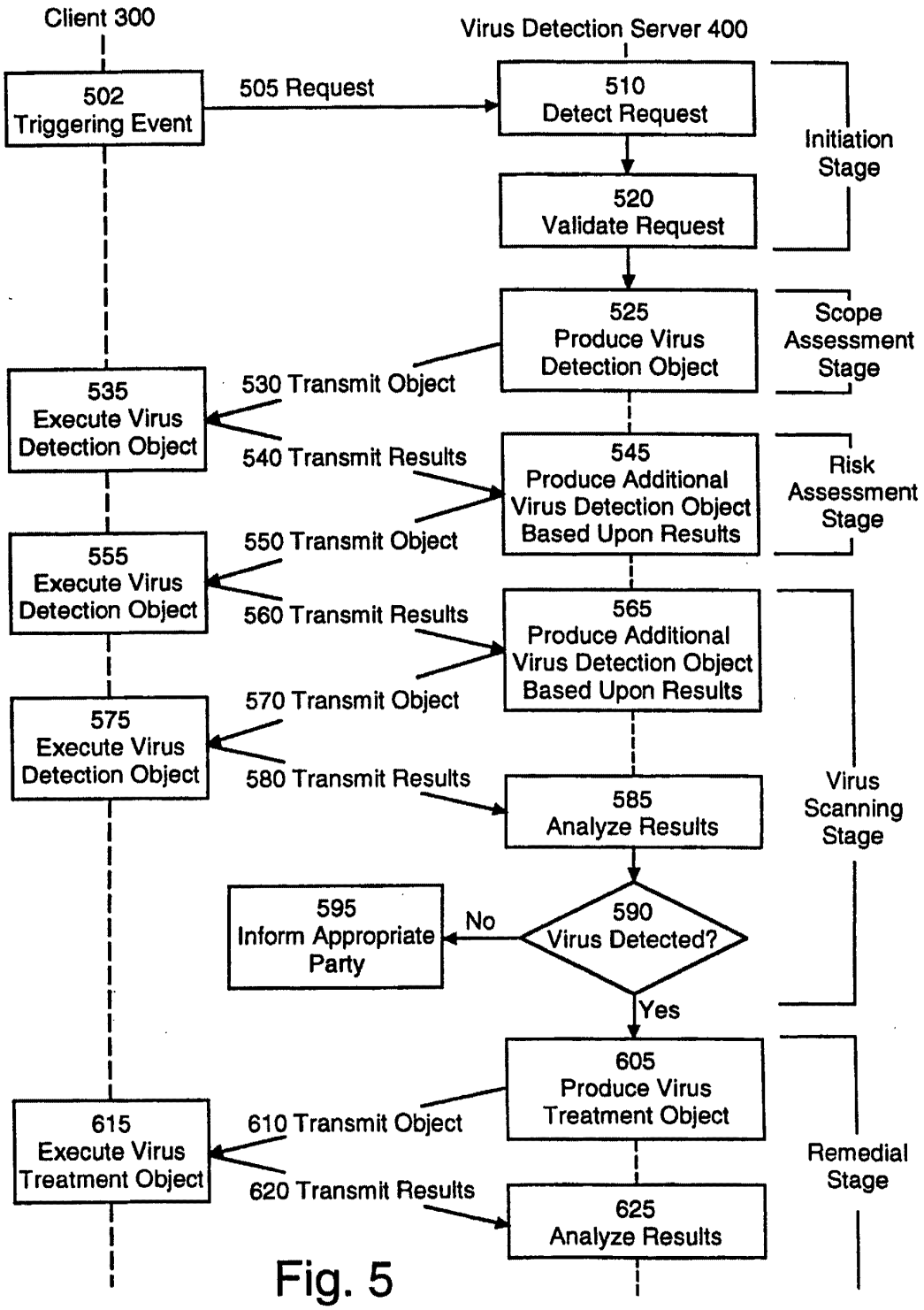

Client 300
Virus Detection Server 400
502 Triggering Event
505 Request
510 Detect Request
520 Validate Request
Initiation Stage
525 Produce Virus Detection Object
Scope Assessment Stage
530 Transmit Object
535 Execute Virus Detection Object
540 Transmit Results
545 Produce Additional Virus Detection Object Based Upon Results
Risk Assessment Stage
550 Transmit Object
555 Execute Virus Detection Object
560 Transmit Results
565 Produce Additional Virus Detection Object Based Upon Results
570 Transmit Object
575 Execute Virus Detection Object
580 Transmit Results
585 Analyze Results
Virus Scanning Stage
595 Inform Appropriate Party
No
590 Virus Detected?
Yes
605 Produce Virus Treatment Object
610 Transmit Object
615 Execute Virus Treatment Object
620 Transmit Results
625 Analyze Results
Remedial Stage

Fig. 5

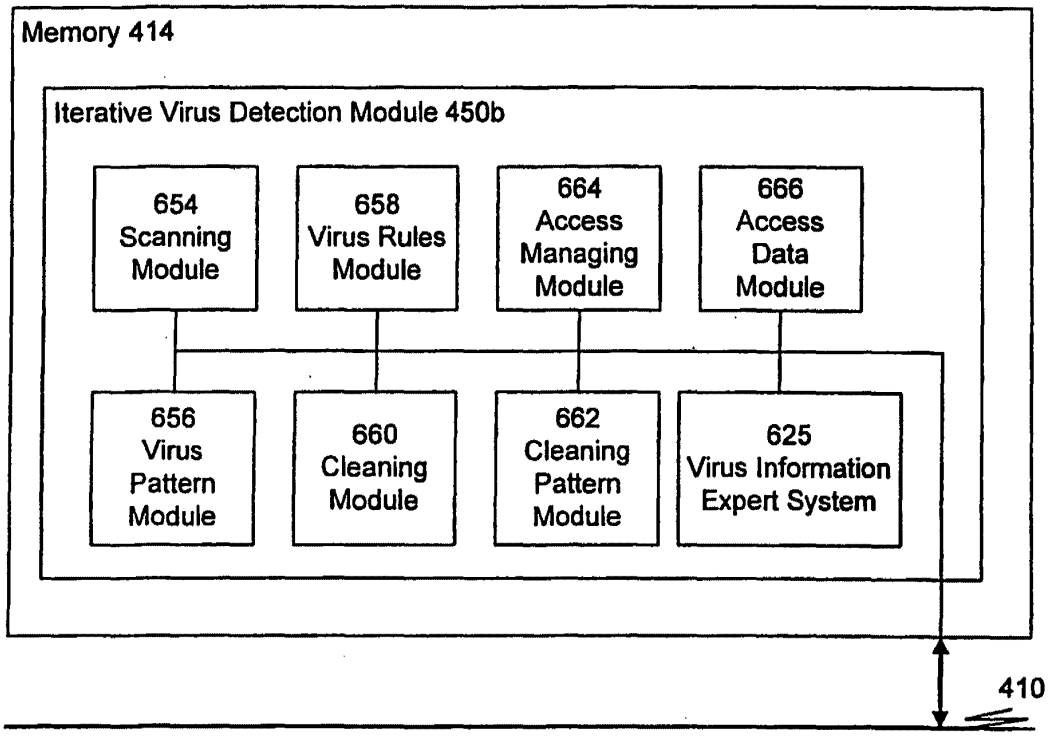

Memory 414
Iterative Virus Detection Module 450b
654 Scanning Module
658 Virus Rules Module
664 Access Managing Module
666 Access Data Module
656 Virus Pattern Module
660 Cleaning Module
662 Cleaning Pattern Module
625 Virus Information Expert System
410

Fig. 6A

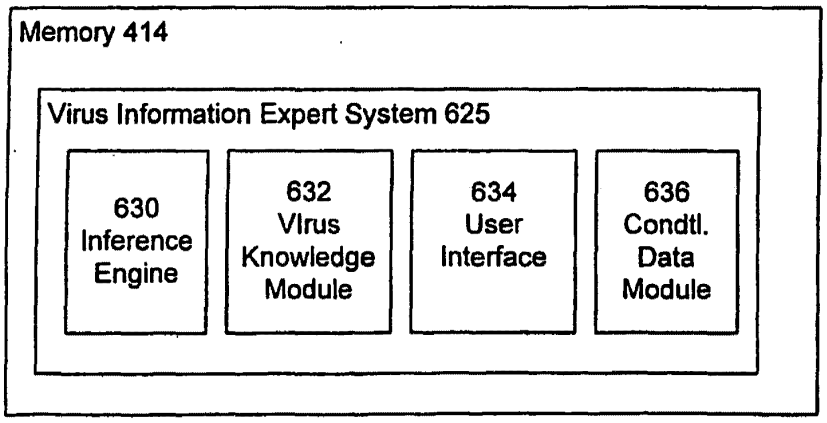

Memory 414
Virus Information Expert System 625
630 Inference Engine
632 Virus Knowledge Module
634 User Interface
636 Condtl. Data Module

Fig. 6B

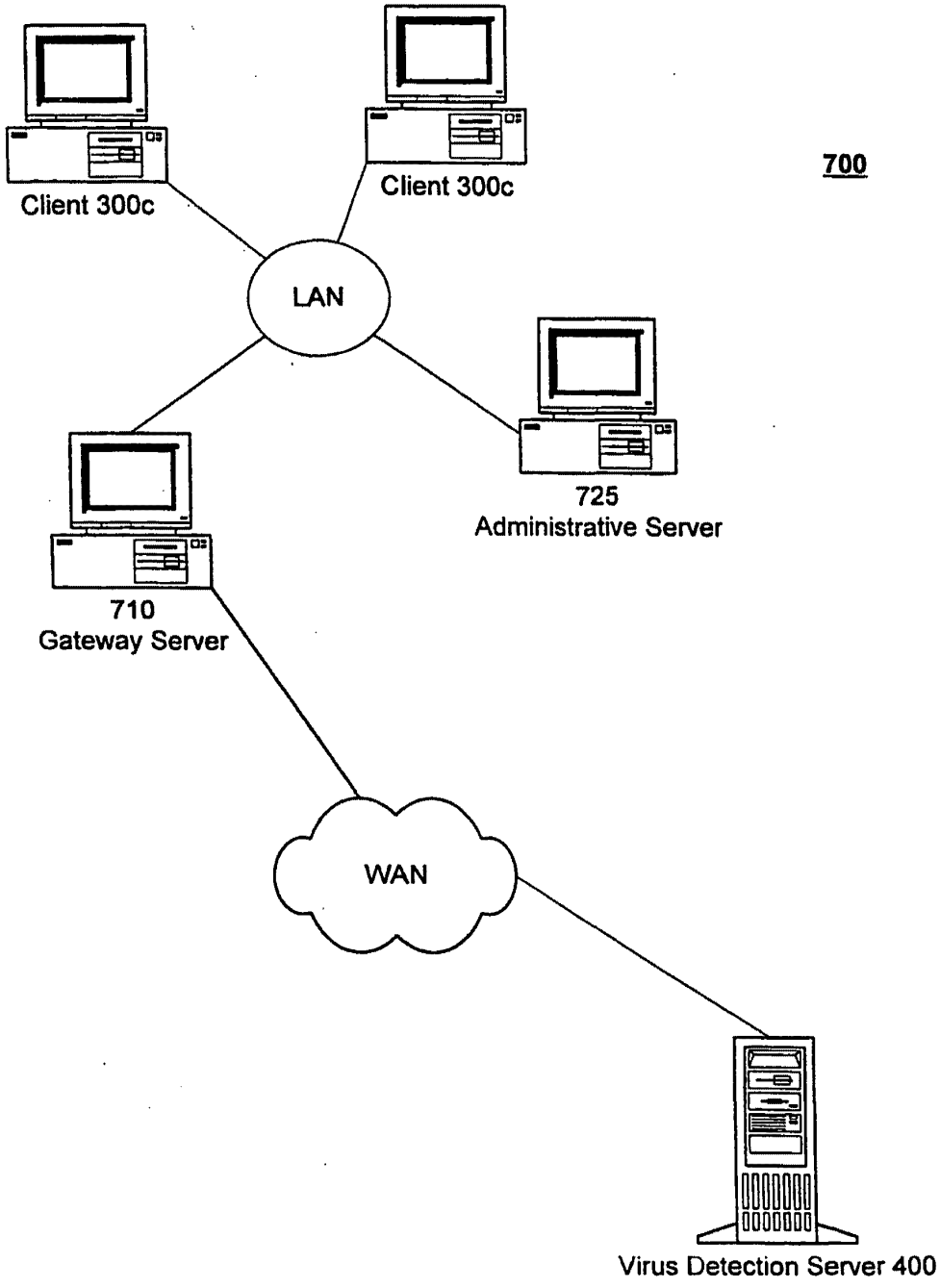

Client 300c
Client 300c
700
LAN
725
Administrative Server
710
Gateway Server
WAN
Virus Detection Server 400

Fig. 7

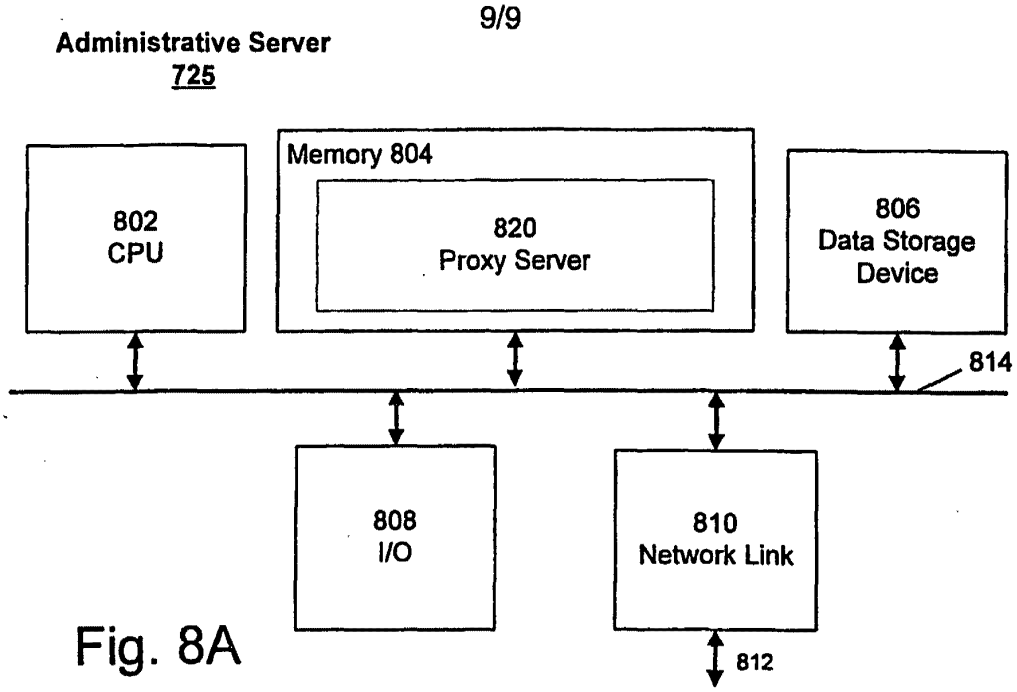

Administrative Server
725
802
CPU
Memory 804
820
Proxy Server
806
Data Storage
Device
814
808
I/O
810
Network Link
812

Fig. 8A

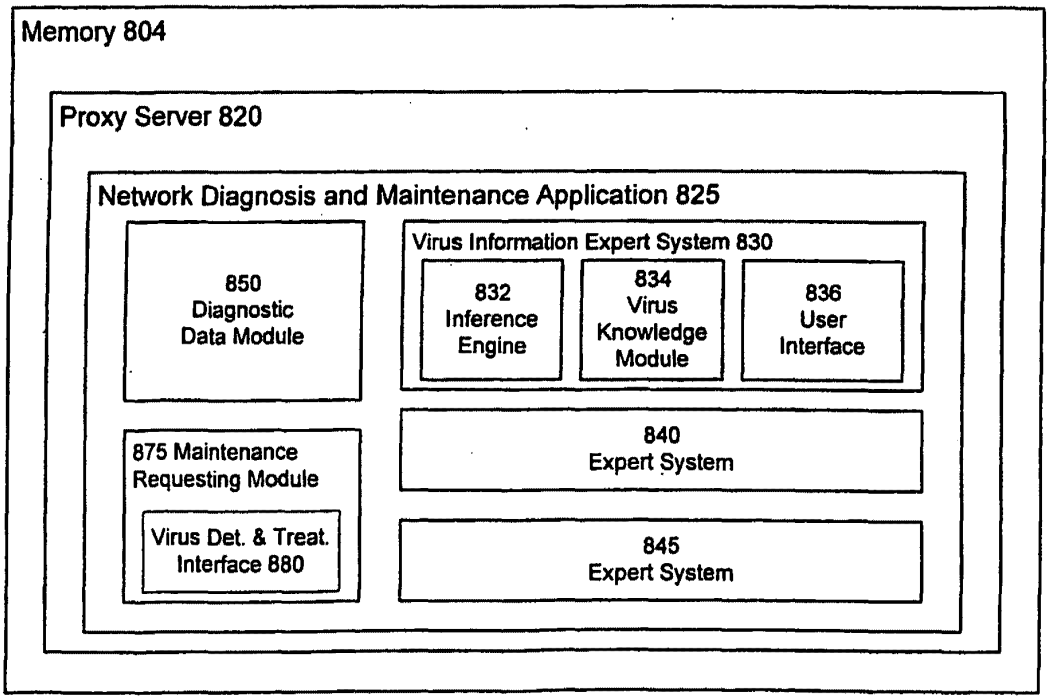

Memory 804
Proxy Server 820
Network Diagnosis and Maintenance Application 825
850
Diagnostic
Data Module
Virus Information Expert System 830
832
Inference
Engine
834
Virus
Knowledge
Module
836
User
Interface
875 Maintenance
Requesting Module
Virus Det. & Treat.
Interface 880
840
Expert System
845
Expert System

Fig. 8B

INTERNATIONAL SEARCH REPORT

Interna...nal Application No
PCT/US 98/03796

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G06F11/00 G06F11/22

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 6 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)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 93 25024 A (CYBERLOCK DATA INTELLIGENCE IN) 9 December 1993 see abstract see page 3, line 5 - page 4, line 16 see page 5, line 23 - column 6, line 15 see page 9, line 3 - line 11 see figure 1 ---	1-3,6, 8-11,13, 15-18
A	US 5 473 769 A (COZZA PAUL D) 5 December 1995 see abstract see column 4, line 31 - column 5, line 15 ---	1,8-10, 15-18
A	US 5 319 776 A (HILE JOHN K ET AL) 7 June 1994 see abstract see column 4, line 59 - column 5, line 21 --- -/--	1,8,10, 15,17

[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
3 June 1998	25.06.98

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	Masche, C

Form PCT/ISA/210 (second sheet) (July 1992)

IPR2024-00027
CrowdStrike Exhibit 1002 Page 976

INTERNATIONAL SEARCH REPORT

Interna...nal Application No
PCT/US 98/03796

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 5 491 791 A (GLOWNY DAVID A ET AL) 13 February 1996 see abstract see column 4, line 21 - line 22 see column 9, line 1 - line 12 see column 10, line 25 - line 38 see column 10, line 54 - column 11, line 2 -----	19,22

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

IPR2024-00027
CrowdStrike Exhibit 1002 Page 977

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 98/03796

Box I Observations where certain claims were found unsearchable (Continuation of item 1 of first sheet)

This International Search Report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the International Application that do not comply with the prescribed requirements to such an extent that no meaningful International Search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box II Observations where unity of invention is lacking (Continuation of item 2 of first sheet)

This International Searching Authority found *multiple* inventions in this international application, as follows:

see additional sheet

1. ☐ As all required additional search fees were timely paid by the applicant, this International Search Report covers all searchable claims.

2. ☒ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this International Search Report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this International Search Report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest.

☐ No protest accompanied the payment of additional search fees.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 978

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. Claims: 1-18

network server iterativly prepares virus detection objects and sends them to client for execution

2. Claims: 19-30

upon request expert system is used to evaluate network diagnostic data and to determine network maintenance

Annex to supplemental sheet b

The International Searching Authority considers that the present application contains 2 inventions. This observation is based on the following reasons:

The prior art has been identified as WO93/25024 (D1).
This document describes (see column 5, line 23 - column 6, line 15) a host system with a security device which scans its file system for virus characteristics and destroys detected virae using information previously downloaded from a remote database at a network monitoring host: if this destruction is unsuccessful further aid is requested from the network monitoring host to detect and destroy the virus. It thus discloses the technical features of a server, supplying virus detecting and destroying maintenance information requested and executed by a client.

Since these technical features are known, they don't make a contribution over the prior art and therefore are not "special technical features" (STF) as defined in rule 13.2 PCT. They cannot be considered to represent the common general inventive concept for the groups of inventions described in claims 1- 18 and 18- 30. The requirements of unity of invention are fullfilled only if for these groups of inventions another common concept, which has to be inventive, can be found. It is the opinion of this ISA that such concept is not present.
Grounds:

1. From the comparison of the first invention and the disclosure of D1, the following technical features of the first invention (claims 1- 18) can be seen to make contribution over D1 and therefore are considered to be "special technical features" (Rule 13.2 PCT) of the first invention:
- receiving results of the execution of the first and second virus detection object, solving the objectivly determined problem of supplying information necessary for tailoring the virus detection objects iterativly and the remedy for the virus.

2. For the same reasoning, from comparison of claims 19- 30 and this same prior art (D1), the "potential" STF of the second invention are:
- an expert system evaluating diagnostic data to reply for a maintenance request, solving the objectiv problem of adapting the maintenance scheme to changing conditions.

The above analysis shows that the STF of the 2 groups of inventions are not the same. A comparison of the objective problems related to the different groups of the inventions, all seen in the light of the description and drawings of the application, shows that they are all different and have no corresponding technical effect. Consequently the STF of the different inventions do not correspond and the requirements of Unity of invention (Rule 13.1 & 13.2 PCT) are not fullfilled.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 980

INTERNATIONAL SEARCH REPORT

Information on patent family members

Interna..ial Application No
PCT/US 98/03796

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9325024 A	09-12-1993	NONE	
US 5473769 A	05-12-1995	US 5502815 A	26-03-1996
		US 5649095 A	15-07-1997
US 5319776 A	07-06-1994	NONE	
US 5491791 A	13-02-1996	NONE	

Form PCT/ISA/210 (patent family annex) (July 1992)

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 1/00 — A1

(11) International Publication Number: **WO 99/53391**

(43) International Publication Date: 21 October 1999 (21.10.99)

(21) International Application Number: PCT/US99/08061

(22) International Filing Date: 14 April 1999 (14.04.99)

(30) Priority Data:
09/060,480 15 April 1998 (15.04.98) US

(71) Applicant: SUN MICROSYSTEMS, INC. [US/US]; 901 San Antonio Road, M/S PAL01-521, Palo Alto, CA 94303 (US).

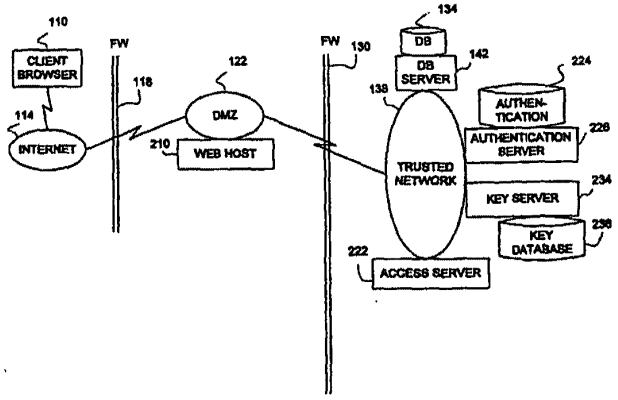
(72) Inventors: ALEGRE, Alfred, A.; 3212 Upper Lock Avenue, Belmont, CA 94002 (US). SHA, Rong, Q.; 1039 Crescent Terrace, Milpitas, CA 95035 (US). SOLEY, William, R.; 1190 Archer Way, Campbell, CA 95008 (US).

(74) Agents: GARRETT, Arthur, S.; Finnegan, Henderson, Farrabow, Garrett & Dunner, L.L.P., 1300 I Street, N.W., Washington, DC 20005-3315 (US) et al.

(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, 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, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, 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), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: APPARATUS AND METHOD FOR TRUSTED NETWORK SECURITY



(57) Abstract

A session key is established for accessing a trusted network from a browser. An authentication process receives identification information from a user at the browser, and authenticates the user by checking the identification information against an authentication database. If the authentication database authenticates the user, a session key is created and stored at the browser. If the user is authenticated, a user profile defining access rights for the user is also retrieved. The user is then presented with access options based on the access rights defined in the user profile. In response to a user selection from the access page, the browser forwards an information request to the trusted network. The request includes a session key. A speaker object processes the information request and session key to form a network request packet. The network request packet is formed in a manner that allows authentication of the speaker object. The session packet is forwarded to a trusted network and processed. The packet is first authenticated to determine if it originated from the speaker object, and then the key is checked for validity at the trusted server. If the key is valid, the information request is processed and the information is returned to the user for display on the browser.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 982

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

APPARATUS AND METHOD FOR TRUSTED NETWORK SECURITY

I. BACKGROUND OF THE INVENTION

A. Field of the Invention

The present invention relates generally to network security, and in particular to apparatus and methods for authenticating a user for allowing access to resources on a trusted network.

B. Description of the Prior Art

Trusted networks provide security limit access to network resources by controlling information passing to, from, and between the resources. For example, information transfer may be controlled by user identification and authentication, access security levels, and physical measures.

Protecting data residing in a company's trusted network is paramount. The most difficult security situations arise when the public is given access to the trusted network, such as through the Internet. Web servers residing between the trusted network and the Internet provide access to databases or legacy applications residing within the trusted network, and may provide unauthorized access to the trusted network from the Internet. Several techniques have been used to make trusted networks more secure from unauthorized access.

Firewalls are one of the most common forms of security. A firewall is a system or structure that limits outside access to a trusted network by limiting the path through which information may flow. For example, whenever the outside web server needs access to the trusted network, the web server submits a request through a firewall port. The port only allows certain protocols, such as HTML, to a specific machine on the trusted network. Firewalls alone are not adequate, however, because they control access based on the location of the user, rather than the identity of the user.

Middleware is also frequently used. Middleware replaces general protocols, such as HTML and SQL, with application-specific protocols. For example, an application issues a request for services in an application-specific form to the middleware residing in the trusted network. The middleware then receives the request and translates it to a general protocol

IPR2024-00027
CrowdStrike Exhibit 1002 Page 984

understood by the server. Intruders, however, can monitor communications between the outside web server and trusted network server, and eventually identify the protocol and patterns of the packets being handled by the middleware. Based on the protocol and patterns, an intruder can access the network to request a service.

To prevent an intruder from monitoring communications, encryption can be incorporated into the architecture. Although effective, encryption does not prevent an intruder from breaching security.

Because no security architecture is 100% secure, multiple security measures are often combined. One approach uses a sub-network that isolates databases from the trusted network. If an intruder gains access into the subnet, the worst that can happen is data residing within the subnet is compromised, but the rest of the trusted network remains secure. This scenario may be adequate in cases where there is no need to interface with other databases or legacy systems within the trusted network.

Fig. 1 is a block diagram showing a typical trusted network security system. The goal of the system is to ensure that resources on trusted network 138 are not improperly accessed by outside entities, such as client browser 110. Access to trusted network 138 is limited in several ways.

Firewall 118 is the first line of defense for providing security to trusted network 138. Firewall 118 may, for example, limit the types of protocol transferred from Internet 114 to DMZ network 122. Web host 126 processes URL requests from client browser 110, and forms a request that is sent over trusted network 138 to database server 142. The request is sent through firewall 130, which provides yet another line of defense. Firewall 130 may also limit the types of information sent by web host 126 to database server 142.

Database server 142 performs a further level of security by insuring that it only processes requests received from web host 126. When web host 126 makes a request, web host 126 also sends a web server identity code with the request. Database server 142 checks the identifier to authenticate that the request is from web host 126. If database server 142 determines that the request is from web host 126, database server 142 retrieves the requested information from database 134, and returns the information to web host 126. Web host 126

IPR2024-00027
CrowdStrike Exhibit 1002 Page 985

transmits the requested information to client browser 110 over DMZ network 122 and Internet 114.

Although firewalls 118 and 130, and authentication of web host 126 by database server 142 provide some security, it is still possible for an intruder to breach security and improperly access resources on the network, such as DB 134. The user at client browser 110 may repeatedly attempt various combinations of access to trusted network 138 until one is found that breaks through the system. Therefore, breaches of security are still possible even with two firewalls and the web server verification performed by database server 142. What is needed then is a higher level of security for trusted network 138 in order to allow access by users on the Internet in a controlled and secure manner.

II. SUMMARY OF THE INVENTION

The present invention relates to trusted networks, and in particular to a method and apparatus for raising security levels of the trusted network.

A system consistent with the present invention comprises a device for processing an original request and key from a requester to form a network request; a device for transferring the network request to a trusted network; a device for processing the network request to extract the key and original request if the request was processed by the device for processing a request and key; and a device for performing the original request if the key is valid.

A method consistent with the present invention comprises processing an original request and key from a requester to form a network request; transferring the network request to a trusted network; processing the network request to extract the key and original request if the request was processed in the step of processing an original request and key; and performing the original request if the key is valid.

Another system for providing access to a resource, consistent with the present invention, comprises a device for storing a key based on requester authentication; a device for forwarding the key to the requester; a device for receiving an original request and the key from the requester; a device for processing the original request and the key from the requester to form a network request; a device for transferring the network request to a trusted network; a device for processing the network request to extract the key if the network request

IPR2024-00027
CrowdStrike Exhibit 1002 Page 986

was processed by the device for processing the original request and the key; and a device for performing the original request if the key is valid.

Another method for providing access to a resource, consistent with the present invention, comprises storing a key based on requester authentication; forwarding the key to the requester; receiving an original request and the key from the requester; processing the original request and the key from the requester to form a network request; transferring the network request to a trusted network; processing the network request to extract the key if the network request was processed by the device for processing the original request and the key; and performing the original request if the key is valid.

The invention overcomes the problems of conventional prior art systems described above. Additional advantages of the invention are apparent from the description which follows, and may be learned by practice of the invention. It is to be understood that both the foregoing general description and the following detailed description are exemplary and explanatory only and are not restrictive of the invention, as claimed.

The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate an embodiment of the invention and together with the description serve to explain the principles of the invention.

III. BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate an embodiment of the invention and, together with the description, serve to explain the advantages and principles of the invention. In the drawings,

Fig. 1 is a block diagram showing a prior art trusted network security system;

Fig. 2 is a block diagram showing an architecture consistent with the principles of the present invention;

Fig. 3 is a block diagram of client browser 110;

Fig. 4 is a flowchart showing the operation of client browser 110 when a user first attempts to access trusted network 138;

Fig. 5 is a block diagram showing web host 210 of Fig. 2;

Fig. 6 is a flowchart showing the processing of web host 210 of Fig. 2;

IPR2024-00027
CrowdStrike Exhibit 1002 Page 987

Fig. 7 is a flowchart showing the user authentication process performed by login process 512;

Fig. 8 is a flowchart showing processing performed by authentication server 226;

Fig. 9 is a block diagram showing in greater detail key server 234 and key database 236;

Fig. 10 is a flowchart showing the operation of client browser 110 when making an access request;

Fig. 11 is a flowchart showing the processing of user requests by speaker object ;

Fig. 12 is a block diagram of access server 222 of Fig. 2;

Fig. 13 is a flowchart showing the processing performed by listener object 1212 of Fig. 12; and

Fig. 14 is a block diagram showing a file security structure that only allows access to particular levels of information.

IV. DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Apparatus and methods consistent with the principles of the present invention create a session key that is stored at a client browser and used to access a trusted network. A session key is created the first time a user requests access to a resource on the trusted network. Subsequently, whenever the user accesses the trusted network during the session in which the session key is made, the session key is transmitted with the access request so that the trusted network can use the session key to authenticate the user. Fig. 2 is a block diagram showing a network system consistent with the principles of the present invention. Client browser 110, Internet 114, firewall 118, DMZ network 122, firewall 130, DB 134, and trusted network 138 have been discussed with respect to Fig. 1. The system of Fig. 2 creates a session key for use in accessing trusted network 138. Creation of the session key will be discussed first, followed by use of the session key in accessing resources of the trusted network.

Creation of a session key is handled primarily by web host 210, authentication server 226 and key server 234. When the user wants to access trusted network 138, the user sends a request from client browser 110, over Internet 114 and DMZ network 122, to web host 210. If this is the first attempt to access trusted network 138, web host 210 must log in the user.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 988

To log in the user, web host 210 requests user authentication information, such as a user ID (UID) and password (PWD), from the user at client browser 110. Upon receiving the UID and PWD, web host 210 requests authentication of the UID and PWD from authentication server 226. Identifying information other than a UID and PWD could also be used.

Authentication server 226 queries authentication database 224 to determine validity of the UID and PWD information. If the UID and PWD are valid, authentication server 226 receives a user access profile from authentication database 224. Authentication server 226 then requests a session key from key server 234. Key server 234 creates a unique and unpredictable session key, and stores the session key, the UID, the PWD, and key expiration criteria in key database 236. Authentication server 226 then transmits the session key and user access profile to web host 210. Web host 210 stores the session key at client browser 110 using a cookie.

Web host 210 also sends trusted network access presentation information to client browser 110. The trusted network access presentation information is created based on the user access profile, and thus includes only selections for accessing resources that the user has access to.

The user selects an access request from the trusted network access presentation information to access trusted network 138. Client browser 110 sends the request, for example a URL associated with the selection, and the session key to web host 210. Web host 210 processes the request and session key to create a network request packet, and transmits the packet to access server 222.

Access server 222 verifies that the network request packet came from web host 210, extracts the session key from the network request packet, and transfers the session key to key server 234 to determine whether it is valid. Key server 234 compares the session key with currently valid session keys to determine if the session key is still valid, and returns the results of the session key validity check to access server 222.

If the session key is still valid, access server 222 performs the request. For example, access server 222 may access DB 134, to obtain requested information from, or write requested information to DB 134. If the request is a read request, access server 222 transmits the requested information to web host 210.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 989

By using a unique and unpredictable session key for each session from the client browser 110, the apparatus and methods consistent with the principles of the present invention provide a high level of security for accesses to trusted network 138. The elements of Fig. 2 will now be discussed in greater detail.

Session Key Creation

Fig. 3 is a block diagram of client browser 110. Client browser 110 is a web browser that requests content, such as HTML pages, over Internet 114, receives the content, and stores the content in memory 318, shown in Fig. 3 as HTML pages 322. Browser 314 also displays the HTML pages on display 310 and stores cookies 324 received over Internet 114. "Cookies" are small files placed on a user's computer by a web site. The cookies allow the web site to maintain state information at client browser 110. For example, cookies allow the web site to collect information on how a user uses the web site. In the present invention, a cookie is used to store the session key at client browser 110 by web host 210.

Fig. 4 is a flowchart showing the operation of client browser 110 the first time a user attempts to access trusted network 138. Browser 314 first displays the home page on display 310 (step 410). In response to a request from the user, browser 314 sends a request, such as a URL, over Internet 114 (step 412), and waits for a response (step 414) by entering a wait state for a period of time (step 416).

If the request is for trusted network 138, web host 210 receives the request and returns a login page to client browser 110 for display (step 418). The login page prompts the user for user authentication information, such as UID and PWD. Browser 314 receives the UID and PWD from the user and forwards the information over Internet 114 to web host 210 (step 420), and goes into a wait state (step 422) for a period of time (step 424). Web server forwards the UID and PWD to authentication server 226 for authentication.

Whether or not UID and PWD are authenticated, web host 210 returns trusted network access presentation information to client browser 110 for display (step 426). If the UID and PWD are not authenticated, the presentation information indicates that authentication was unsuccessful. If UID and PWD are authenticated, however, web host 210, sends trusted network access presentation information and a cookie with the session key and browser 314 stores the cookie (step 428). The trusted network access presentation information has a menu of selections for accessing resources on trusted network 138.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 990

Fig. 5 is a block diagram showing web host 210 of Fig. 2. Web host 210 is comprised of session manager 510, login process 512, and speaker object 514. Web host 210 may be implemented as a typical network server comprised of hardware and software. Session manager 510, login process 512, and speaker object 514 represent software modules running on the server. Session manager 510 is responsible for coordinating the operations of login process 512 and speaker object 514. Additionally, session manager 510 interfaces with DMZ network 122 for receiving information from and transmitting information to DMZ network 122.

Fig. 6 is a flowchart showing the processing of web host 210 of Fig. 2. Web host 210 first receives a request from client browser 110 (step 610). If the request contains a cookie (step 612: YES) the request is forwarded to speaker object 514 (step 614). If the request does not contain a cookie (step 612: NO) the request is checked for UID and PWD information (step 616).

If the UID and PWD information is present (step 616: YES) session manager 510 forwards the UID and PWD to login process 512 (step 620). If there is no UID and PWD information (step 616: NO), web server returns a login page to client browser 110 (step 618) to prompt the user for login information.

Fig. 7 is a flowchart showing the user authentication process performed by login process 512. Login process 512 first receives user authentication information, such as a UID and PWD, from client browser 110, and forwards the authentication information to authentication server 226 via session manager 510 (step 710), and enters a wait state (step 712) to wait for the results of the authentication process.

If authentication server 226 returns a negative authentication (step 714: NO), meaning that the UID and PWD were not authenticated, login process 512 causes session manager 510 to return an unauthorized access page to client browser 110 (step 716). If UID and PWD are authenticated (step 714: YES), login process 512 receives a session key and user profile from authentication server 226. Login process 512 creates a trusted network access menu page using the user profile and a cookie containing the session key, and forwards the page and cookie to client browser 110 (step 718).

Fig. 8 is a flowchart showing the processing performed by authentication server 226. Authentication server 226 first receives the UID and PWD from login process 512 as part of

IPR2024-00027
CrowdStrike Exhibit 1002 Page 991

the initial login by the user at client browser 110 (step 810). Authentication server 226 queries authentication database 224 to determine if the UID and PWD are valid (step 812). Authentication database 224 stores information defining which users may access resources on trusted network 138. Authentication database 224 also stores user profile information that defines the types of access each user has to the resources on trusted network 138. On the one hand, if the UID and PWD are not valid (step 814: NO), authentication server 226 returns an invalid status to login process 512 (step 816). On the other hand, if authentication server 226 receives an indication from authentication database 224 that UID and PWD are valid (step 814: YES), a session key is requested from key server 234 (step 818). Key server 234 creates a new key, stores it in key database 236 with the UID and PWD, and expiration criteria.

In addition to receiving a validation indication from authentication database 224, authentication server 226 also receives a user profile that specifies the user's access rights to trusted network 138. After receiving a new session key from key server 234, authentication server 226 returns an indication of the UID and PWD being valid, session key, and the user profile to login process 512 (step 820).

Fig. 9 is a block diagram showing key server 234 and key database 236. Key server may be implemented using a processor and memory. Key server 234 includes key manager 910 and key creator 912. Key manager 910 receives the session key request, along with the UID and PWD, from authentication server 226, and requests key creation by key creator 912. The session key must be unique, unpredictable, and from a sufficiently large number space so that it is infeasible to guess its value. For example, a 64-bit or larger random number encoded in Base64 may be used. Upon receiving the new session key from key creator 912, key manager 910 stores the session key, UID, and PWD, along with expiration criteria for the session key, in key database 236. Key manager 910, then transfers the new session key to authentication server 226 for transmission to login manager 214. Key manager 910 also monitors the expiration criteria, and deletes keys in accordance with the criteria, or when the user voluntarily logs out.

Use of Session Key

IPR2024-00027
CrowdStrike Exhibit 1002 Page 992

Fig. 10 is a flowchart showing the operation of client browser 110 accessing trusted network 138 after the session key has been stored in client browser 110. The user makes a selection from the trusted network access web page created by login process 512 using the user profile. Browser 314 receives the request (step 1010), and retrieves the cookie holding the session key (step 1012). Browser 314 then forwards the request and cookie to web host 210 (step 1014), and goes into a wait state to wait for the requested information from web host 210 (step 1016). When browser 314 receives the requested content, for example in the form of an HTML page, the content is displayed (step 1018). If the user does not log off (step 1020: NO) the process resumes with receiving another user request. If the user chooses to log off (step 1020: YES), the client browser forwards a logout request to the web host which instructs both the browser and the key server to delete the session key.

Fig. 11 shows processing of user requests to access trusted network 138 by speaker object 514. Speaker object 514 first receives the action request and session key from client browser 110 (step 1110). In response, speaker object 514 creates a request packet that includes the action request and the session key, sends the packet to access server 222 (step 1112), and goes into a wait state (step 1114) until a response is received from access server 222.

The packet created by speaker object 514 may be created in a variety of ways. For example, the packet may be created by merely concatenating a web server identifier, speaker object identifier, or other identifier, to the session key and URL request received from the user. Alternatively, speaker object 514 may sign the packet by encrypting it with the private key. The private key could be pre-programmed at speaker object 514, or may be received from trusted network 138 (not shown). The packet is created in such a way that when listener object 230 receives the packet, the packet can be identified as originating from speaker object 514.

When speaker object 514 receives a response to the request from access server 222 (step 1116), speaker object 514 creates a web page based on the response, and sends the web page to session manager 510 for transmission to client browser 110 (step 1118).

Fig. 12 is a block diagram showing access server 222 of Fig. 2. Access server 222 is comprised of access manager 1210 and listener object 1212. Access server may be implemented as a network server as is well understood in the art. For example, access server

IPR2024-00027
CrowdStrike Exhibit 1002 Page 993

222 may be implemented using a processor and memory. Access manager 1210 interfaces access server 222 with trusted network 138, and manages the operations performed by listener object 1212. Listener object 1212 is responsible for receiving network requests from speaker object 514, verifying the validity of the request, and performing the request if valid.

Fig. 13 is a flowchart showing the processing performed by listener object 1212. Listener object 1212 first receives the network request packet containing the action request and session key from speaker object 514 (step 1310). For example, if speaker object 514 signed the packet using speaker identification information, listener object must verify the signature before extracting the action request and session key. If the packet is not from speaker object 514, the signature verification process by listener object 1212 fails and the key and request sent from speaker object 514 will be rejected by listener object 1212.

Alternatively, speaker object 514 and listener object 1212 may each have the same web server identification information. Speaker object 514 attaches the web server identification to the packet, which is verified by listener object 1212 upon receipt. Listener object 1212 processes the packet to extract the session key and network access request from the packet (step 1311), sends the key to key server 234 for validation (step 1312), and goes into a wait state step 1314 to wait for the validation of the session key (step 1314).

If key server 234 determines that the session key received from listener object 1212 is invalid (step 1316: NO), listener object 1212 returns an invalid status indication to speaker object 514 (step 1118). If key server 234 returns a UID and access profile and the network access request received from speaker object 514 is permitted by the access profile (step 1316: YES), listener object 1212 performs the network access request (step 1320). In addition to the check against the access profile received from key server 234, applications requiring extra fine grained access control may use the UID received from key server 234 in combination with a local data base of access rules (not shown) to implement additional access control policies.

The request may include one or more requests for operations by resources on network 138. Performing the request may, for example, include a read operation on DB 134. After performing the request, listener object 1212 returns the requested data (if any) and status of the request to speaker object 514 (step 1322).

IPR2024-00027
CrowdStrike Exhibit 1002 Page 994

Fig. 14 is a block diagram showing a file security structure that may be used to limit user access to particular levels of information. Fig. 14, for example, may represent the data structure of DB 134. A user associated with a particular UID, PWD, or session key be restricted to only accessing particular levels of data. For example, a particular UID may only allow access to level three and level four information of Fig. 12. The level access information is included in the user profile used by login process 512 to create the network access page presented to the user at client browser 110, as discussed above. This allows trusted network 138 to have varying degrees of protection depending on the particular user at client browser 110. The user profile may also define a variety of other types of acceptable operations for a user. The profile may define, for example, which resource(s) the user can use, and the operations that can be performed for the user by the resource.

Conclusion

It will be apparent to those skilled in the art that various modifications and variations can be made in the network access apparatus and methods consistent with the principles of the present invention without departing from the scope or spirit of the invention. Although a preferred embodiment has been described above, other variations are possible within the spirit and scope consistent with the principles of the present invention.

For example, speaker object 512 could be located in the client browser 110 by using Java applets to implement the speaker object on client browser 110. The user profile stored with the authentication information in authentication database 224 may include access levels corresponding to each particular user. For example, with respect to Fig. 12, the user having access to levels 3 and 4 will have a user profile indicating that the user can only be presented with URL selections allowing access to levels 3 and 4. Therefore, when the user profile is sent to login process 512, login process 512 creates a user access web page that only allows request to levels 3 and 4.

Other embodiments of the invention will be apparent to those skilled in the art from consideration of the specification and practice of the disclosed embodiments. The specification and examples are exemplary only, and the true scope and spirit of the invention is defined by the following claims and their equivalents.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 995

WE CLAIM:

1. A system for performing a request, comprising:

means for processing an original request and key from a requester to form a network request;

means for transferring the network request to a trusted network;

means for processing the network request to extract the key and original request if the request was processed by the means for processing a request and key; and

means for performing the original request if the key is valid.

2. The system according the claim 1, wherein the means for processing comprises:

means for including in the network request information identifying the means for transferring.

3. The system according to claim 1, wherein the means for processing comprises:

means for signing information to form the network request.

4. The system according to claim 1, further comprising:

means for storing key information based on authentication of requester identification information; and

means for determining validity of the key based on the key information.

5. The system according to claim 1, wherein the means for performing the original request comprises:

means for accessing a trusted network resource to perform the original request.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 996

6. A method for performing a request, comprising:

processing an original request and key from a requester to form a network request;

transferring the network request to a trusted network;

processing the network request to extract the key and original request if the request was processed in the step of processing an original request and key; and

performing the original request if the key is valid.

7. The method according the claim 6, wherein the step of processing an original request and key includes the substep of:

including intermediate transferor information in the network request.

8. The method according to claim 6, wherein the step of processing an original request and key includes the substep of:

signing information to form the network request.

9. The method according to claim 6, further including the steps of:

storing key information based on authentication of requester identification information; and

determining validity of the key based on the key information.

10. The method according to claim 6, wherein the step of performing the original request comprises:

accessing a trusted network resource to perform the original request.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 997

11. A computer program product comprising:

a computer usable medium having computer readable code embodied therein for performing a request, the computer usable medium comprising:

a processing module configured to process an original request and key from a requester to form a network request;

a transferring module configured to transfer the network request to a trusted network;

a processing module configured to process the network request to extract the key and original request if the request was processed by the means for processing a request and key; and

a performing module configured to perform the original request if the key is valid.

12. The computer program product of claim 11, wherein the processing module configured to process an original request and key comprises:

an including module configured to include in the network request information identifying the means for transferring.

13. The computer program product of claim 11, wherein the processing module configured to process an original request and key comprises:

a signing module for signing information to form the network request.

14. The computer program product of claim 11, further comprising:

a storing module configured to store key information based on authentication of requester identification information; and

a determining module configured to determine validity of the key based on the key information.

15. The computer program product of claim 11, wherein the performing module comprises:

an accessing module configured to access a trusted network resource to perform the original request.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 998

16. A system for providing access to a resource, comprising:

means for storing a key based on requester authentication;

means for forwarding the key to the requester;

means for receiving an original request and the key from the requester;

means for processing the original request and the key from the requester to form a network request;

means for transferring the network request to a trusted network;

means for processing the network request to extract the key if the network request was processed by the means for processing the original request and the key; and

means for performing the original request if the key is valid.

17. The system according to claim 16, further comprising:

means for authenticating requester identification information; and

means for creating the key in response to authentication of requester identification information.

18. The system according to claim 16, further comprising:

means for authenticating requester information; and

means for forwarding requester access profile information based on the authentication.

19. The system according to claim 16, further comprising:

means for receiving user access profile information;

means for creating display information having user access options based on the user access profile information; and

means for forwarding the display information to the requester.

20. The system according the claim 16, further comprising:

means for consulting the means for storing a key to determine validity of the extracted key.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 999

21. The system according to claim 16,

wherein the means for processing the original request and the key comprises means for including information identifying the means for transferring; and

wherein the means for processing the network request comprises means for determining if the information identifying the means for transferring matches predetermined criteria.

22. The system according to claim 16

wherein the means for processing the original request and the key comprises means for signing first information to form the network request; and

wherein the means for processing the network request comprises means for verifying the network request to derive the first information.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1000

23. A method for providing access to a resource, comprising:

storing a key based on requester authentication;

forwarding the key to the requester;

receiving an original request and the key from the requester;

processing the original request and the key from the requester to form a network request;

transferring the network request to a trusted network;

processing the network request to extract the key if the network request was processed by the step of processing the original request and the key; and

performing the original request if the key is valid.

24. The method according to claim 23, further including the steps of:

authenticating requester identification information; and

creating the key in response to authentication of requester identification information.

25. The method according to claim 23, further including the steps of:

authenticating requester information; and

forwarding requester access profile information based on the authentication.

26. The method according to claim 23, further including the steps of:

receiving user access profile information;

creating display information having user access options based on the user access profile information; and

forwarding the display information to the requester.

27. The method according the claim 23, further including the step of:

determining validity of the extracted key.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1001

28. The method according to claim 23,

wherein the step of processing the original request and the key includes a substep of including information identifying a means for transferring the network request; and

wherein the step of processing the network request includes a substep of determining if the information identifying a means for transferring matches predetermined criteria.

29. The method according to claim 23,

wherein the step of processing the original request and the key includes a substep of signing first information to form the network request; and

wherein the step of processing the network request includes a substep of verifying the network request to derive the first information.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1002

30. A computer program product comprising:

a computer usable medium having computer readable code embodied therein for providing access to a resource, the computer usable medium comprising:

a storing module configured to store a key based on requester authentication;

a forwarding module configured to forward the key to the requester;

a receiving module configured to receive an original request and the key from the requester;

a processing module configured to process the original request and the key from the requester to form a network request;

a transferring module configured to transfer the network request to a trusted network;

a processing module configured to process the network request to extract the key if the network request was processed by the step of processing the original request and the key; and

a performing module configured to perform the original request if the key is valid.

31. The computer program product of claim 30, further comprising:

an authenticating module configured to authenticate requester identification information; and

a creating module configured to create the key in response to authentication of requester identification information.

32. The computer program product according to claim 30, further comprising:

an authenticating module configured to authenticate requester information; and

a forwarding module configured to forward requester access profile information based on the authentication.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1003

33. The computer program product of claim 30, further comprising:

a receiving module configured to receive user access profile information;

a creating module configured to display information having user access options based on the user access profile information; and

a forwarding module configured to forward the display information to the requester.

34. The computer program product of claim 30, further comprising:

a consulting module configured to consult the storing module configured to store a key to determine validity of the extracted key.

35. The computer program product of claim 30,

wherein the processing module configured to process the original request and the key comprises an including module configured to include information in the network request identifying the transferring module; and

wherein the processing module configured to process the network request comprises a determining module configured to determine if the information identifying a means for transferring matches predetermined criteria.

36. The computer program product according to claim 30,

wherein the processing module configured to process the original request and the key comprises an encryption module configured to sign first information to form the network request; and

wherein the processing module configured to process the network request comprises a decryption module configured to verify the network request to derive the first information.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1004

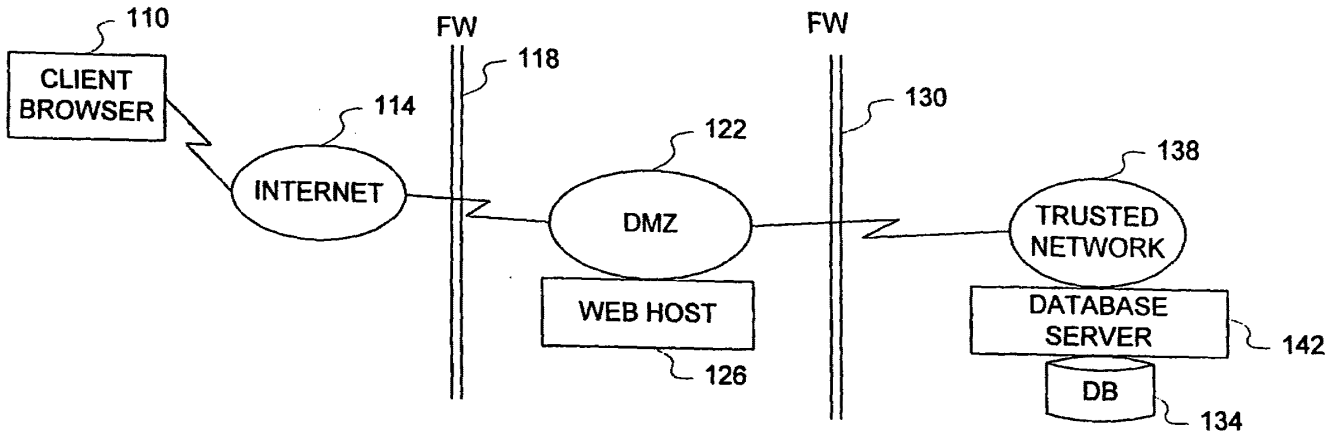


FIG. 1

PRIOR ART

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1005

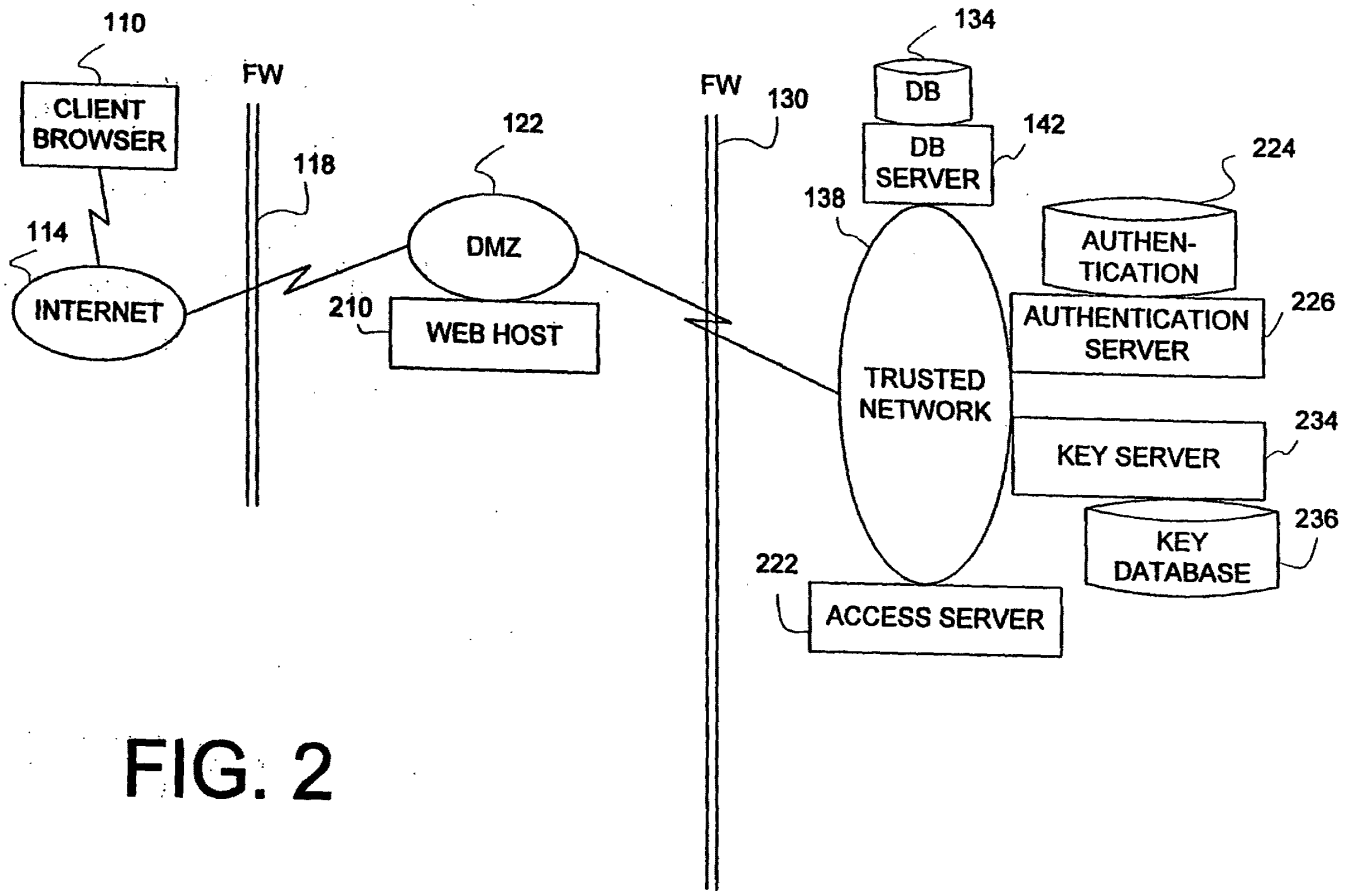

110
CLIENT BROWSER
114
INTERNET
FW
118
122
DMZ
210
WEB HOST
FW
130
134
DB
142
DB SERVER
138
TRUSTED NETWORK
224
AUTHEN-TICATION
226
AUTHENTICATION SERVER
234
KEY SERVER
236
KEY DATABASE
222
ACCESS SERVER

FIG. 2

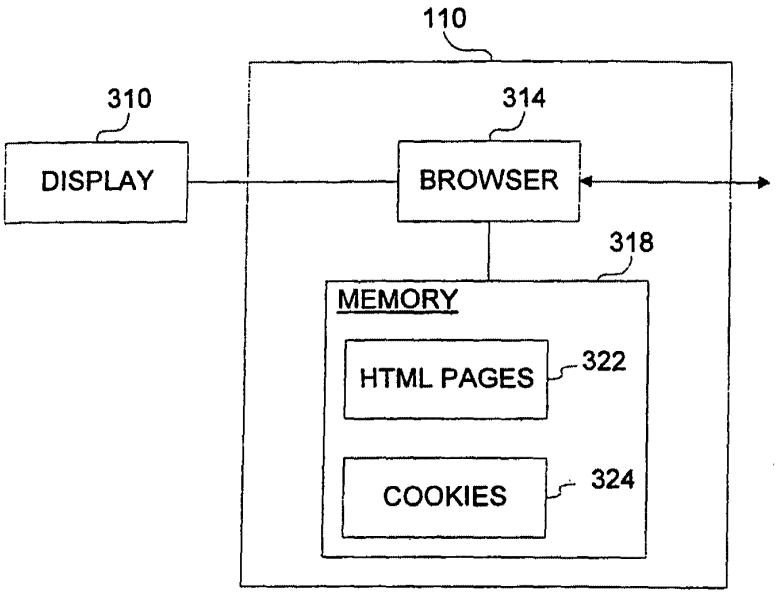

110
310
DISPLAY
314
BROWSER
318
MEMORY
HTML PAGES
322
COOKIES
324

FIG. 3

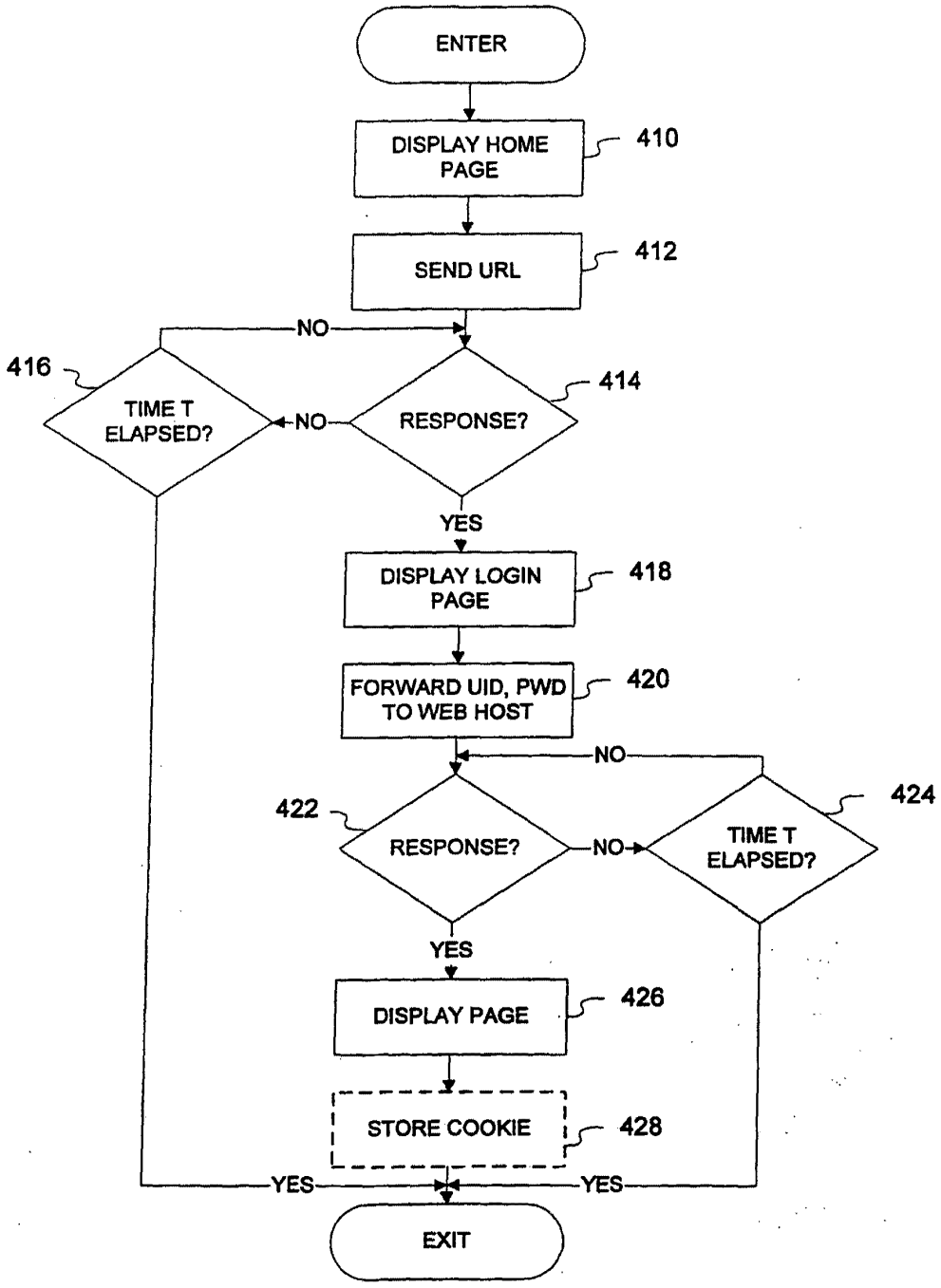

ENTER
DISPLAY HOME PAGE
410
SEND URL
412
NO
416
TIME T ELAPSED?
NO
RESPONSE?
414
YES
DISPLAY LOGIN PAGE
418
FORWARD UID, PWD TO WEB HOST
420
NO
422
RESPONSE?
NO
TIME T ELAPSED?
424
YES
DISPLAY PAGE
426
STORE COOKIE
428
YES
YES
EXIT

FIG. 4

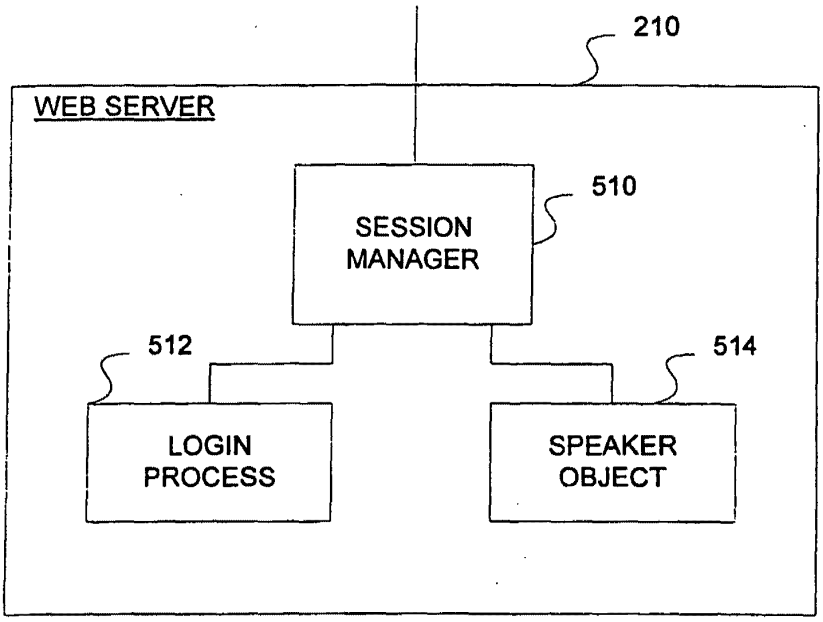

210
WEB SERVER
510
SESSION MANAGER
512
514
LOGIN PROCESS
SPEAKER OBJECT

FIG. 5

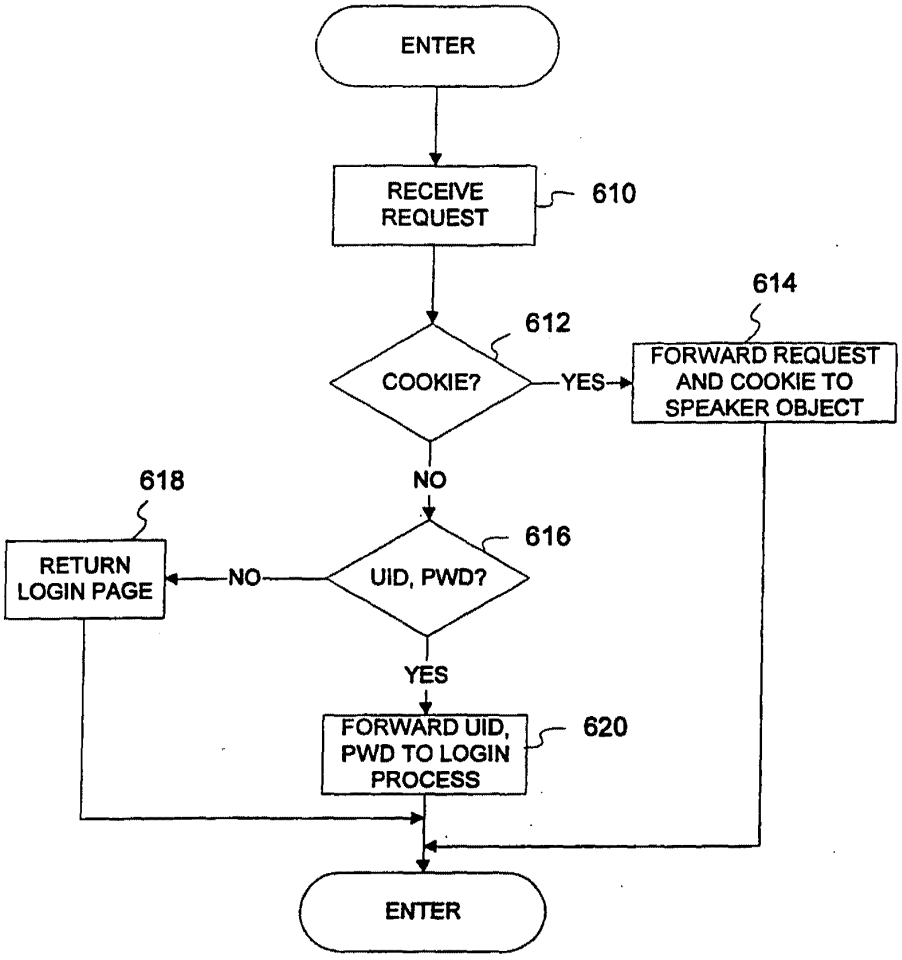

ENTER
RECEIVE REQUEST
610
612
614
COOKIE?
YES
FORWARD REQUEST AND COOKIE TO SPEAKER OBJECT
NO
618
616
RETURN LOGIN PAGE
NO
UID, PWD?
YES
FORWARD UID, PWD TO LOGIN PROCESS
620
ENTER

FIG. 6

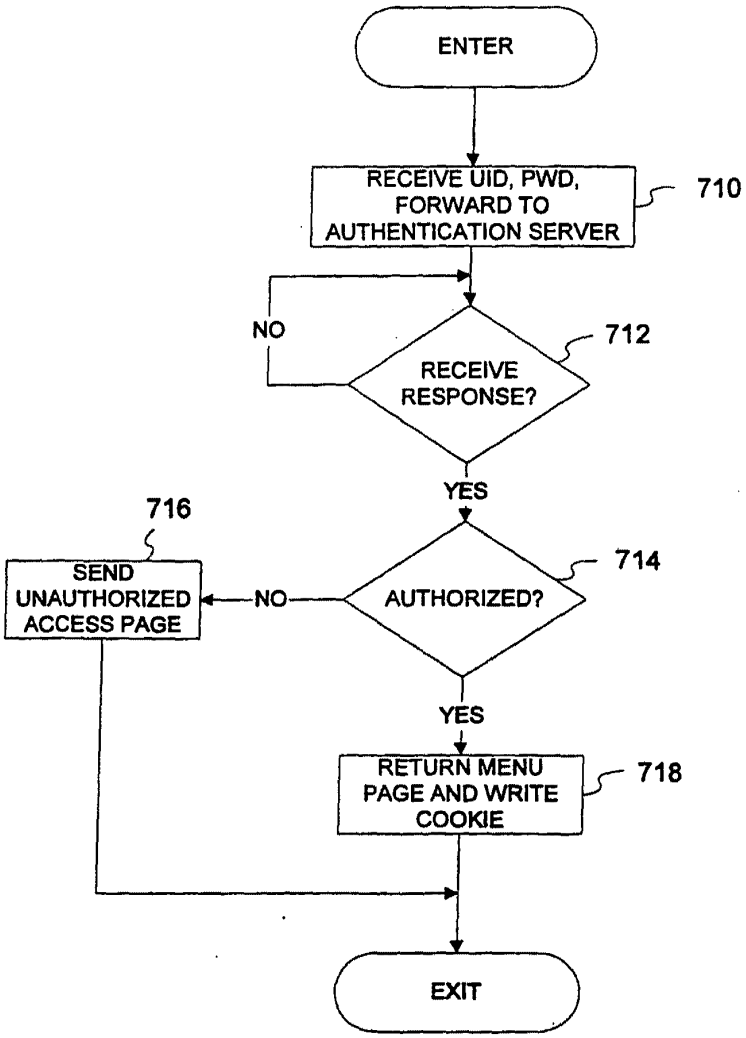

ENTER
RECEIVE UID, PWD, FORWARD TO AUTHENTICATION SERVER
710
NO
RECEIVE RESPONSE?
712
YES
716
SEND UNAUTHORIZED ACCESS PAGE
NO
AUTHORIZED?
714
YES
RETURN MENU PAGE AND WRITE COOKIE
718
EXIT

FIG. 7

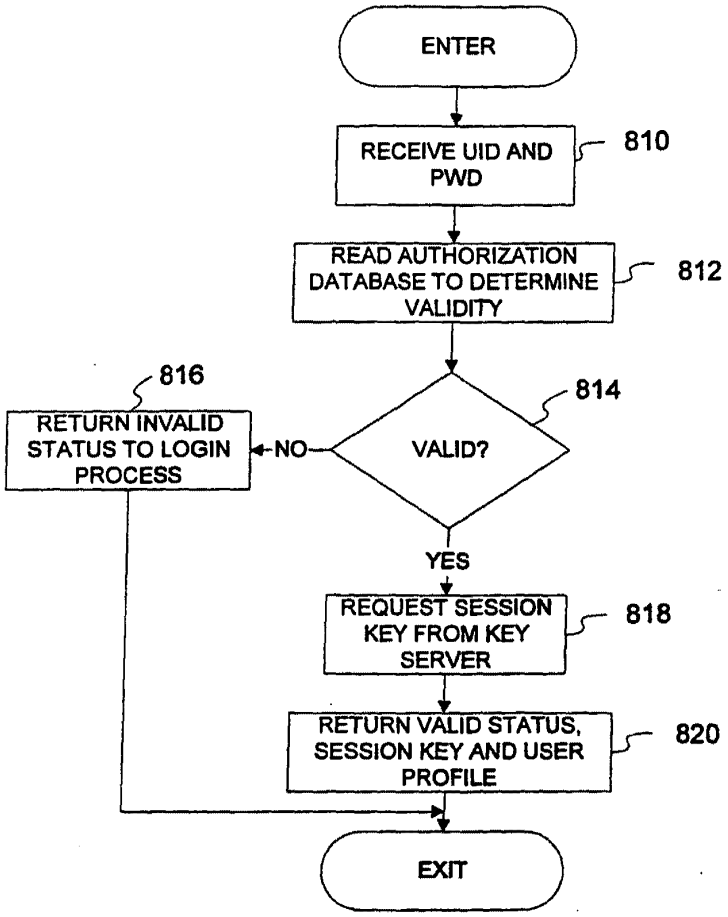

ENTER
RECEIVE UID AND PWD
810
READ AUTHORIZATION DATABASE TO DETERMINE VALIDITY
812
VALID?
814
NO
RETURN INVALID STATUS TO LOGIN PROCESS
816
YES
REQUEST SESSION KEY FROM KEY SERVER
818
RETURN VALID STATUS, SESSION KEY AND USER PROFILE
820
EXIT

FIG. 8

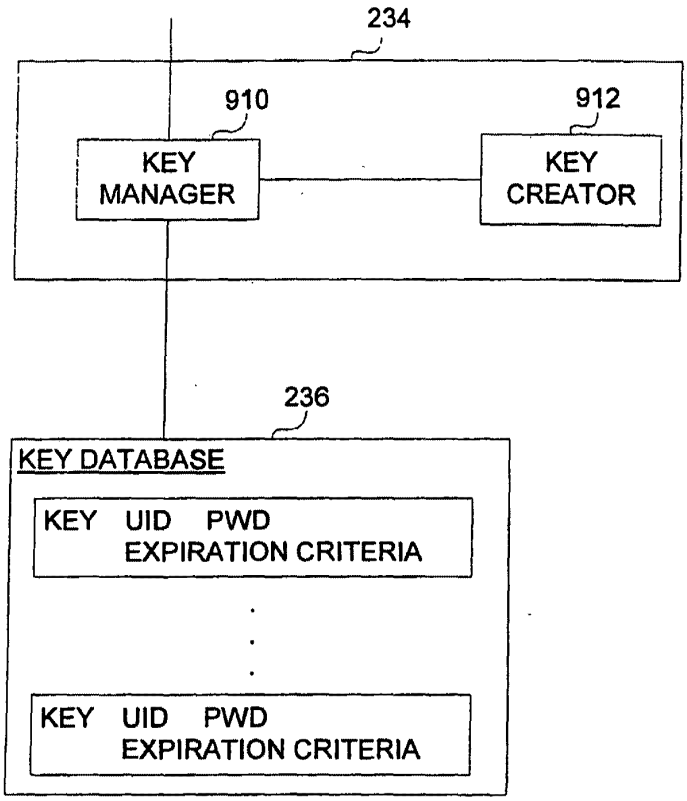

234
910
912
KEY MANAGER
KEY CREATOR
236
KEY DATABASE
KEY UID PWD EXPIRATION CRITERIA
KEY UID PWD EXPIRATION CRITERIA

FIG. 9

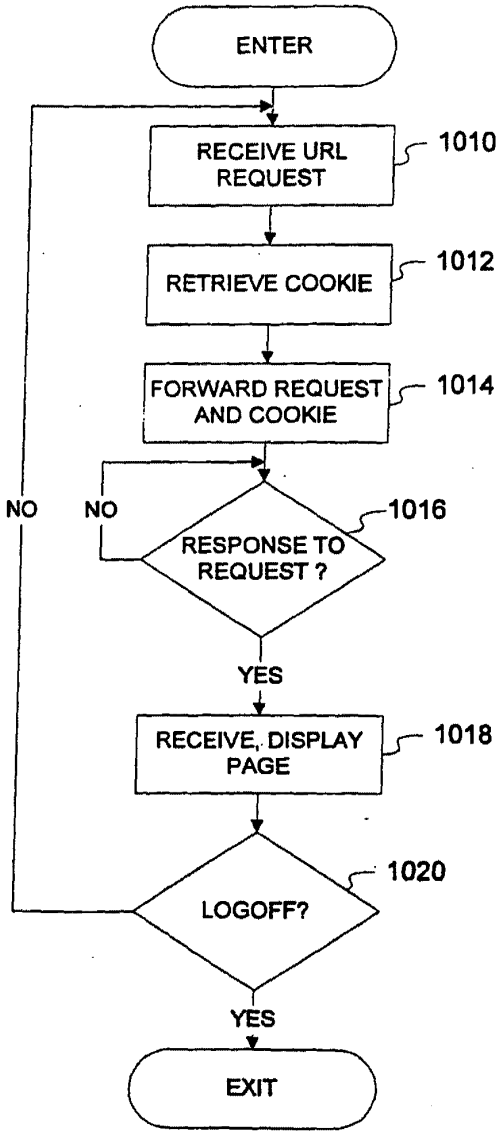

ENTER
RECEIVE URL REQUEST
1010
RETRIEVE COOKIE
1012
FORWARD REQUEST AND COOKIE
1014
NO
NO
RESPONSE TO REQUEST ?
1016
YES
RECEIVE, DISPLAY PAGE
1018
LOGOFF?
1020
YES
EXIT

FIG. 10

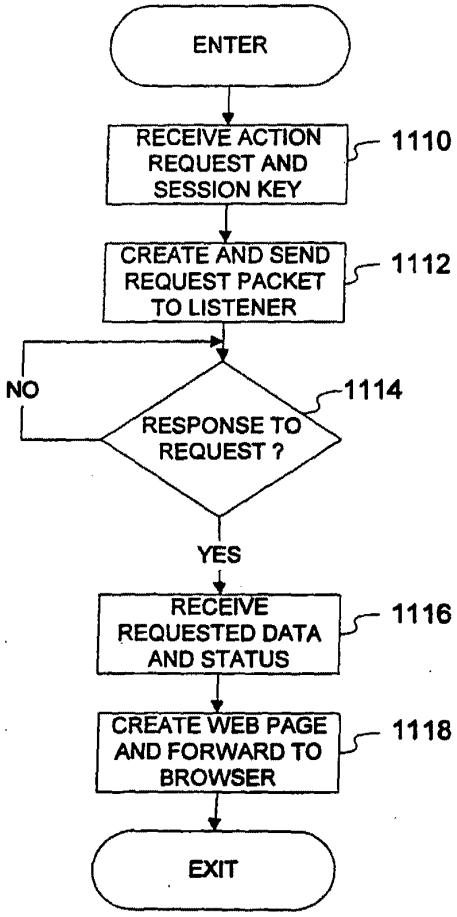

ENTER
RECEIVE ACTION REQUEST AND SESSION KEY
1110
CREATE AND SEND REQUEST PACKET TO LISTENER
1112
NO
RESPONSE TO REQUEST ?
1114
YES
RECEIVE REQUESTED DATA AND STATUS
1116
CREATE WEB PAGE AND FORWARD TO BROWSER
1118
EXIT

FIG. 11

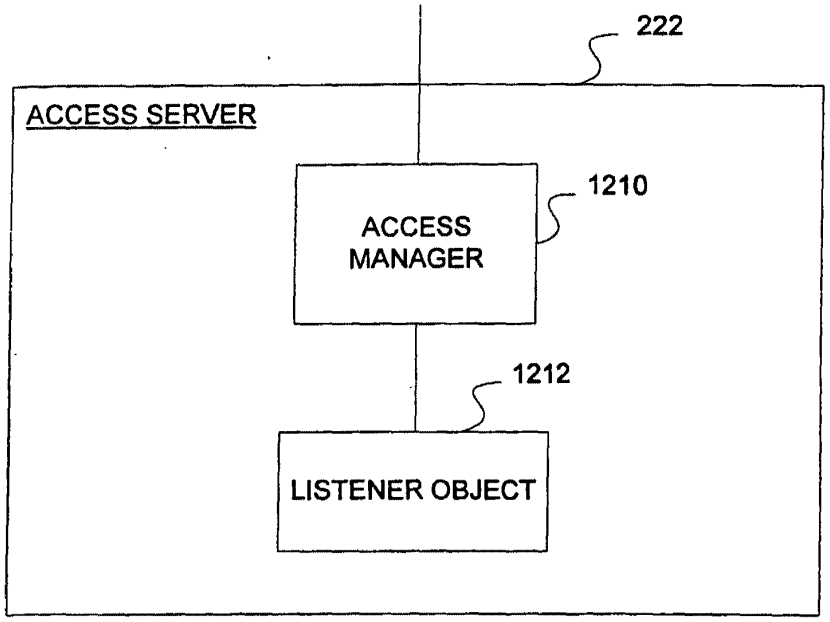

222
ACCESS SERVER
1210
ACCESS MANAGER
1212
LISTENER OBJECT

FIG. 12

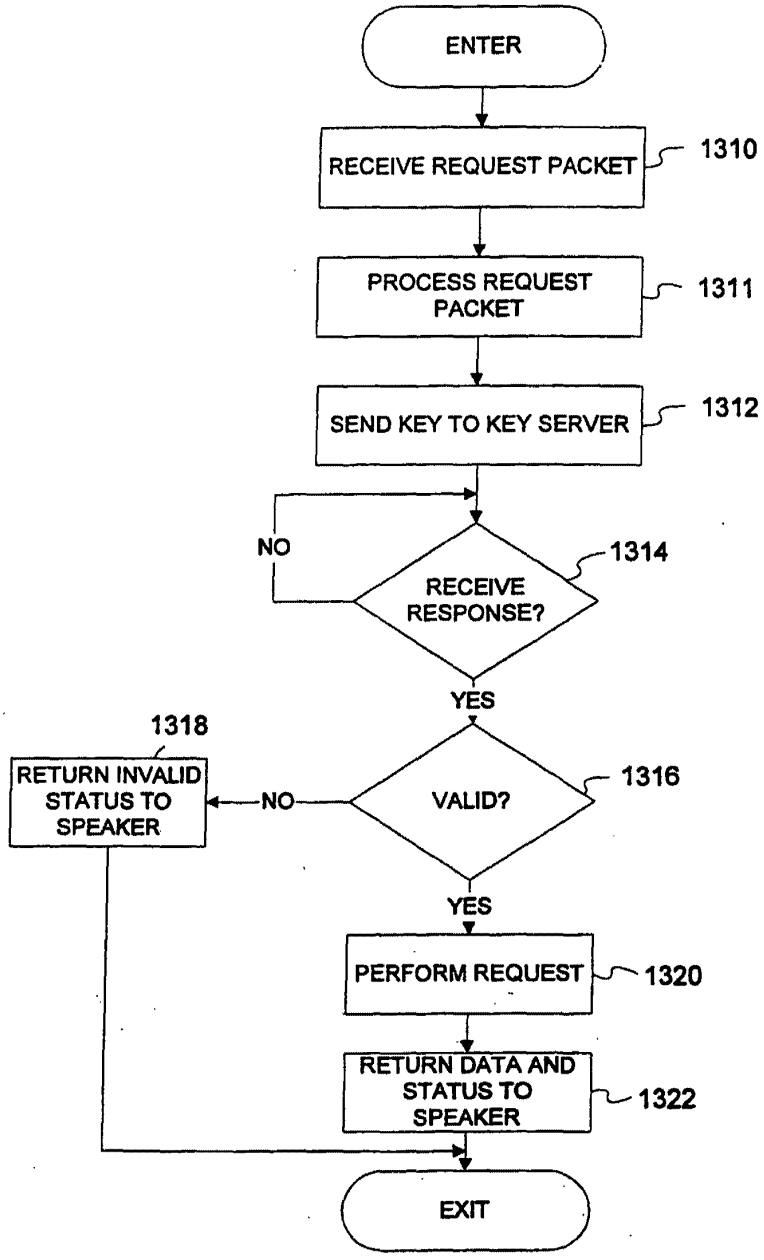

ENTER
RECEIVE REQUEST PACKET
1310
PROCESS REQUEST PACKET
1311
SEND KEY TO KEY SERVER
1312
NO
RECEIVE RESPONSE?
1314
YES
1318
RETURN INVALID STATUS TO SPEAKER
NO
VALID?
1316
YES
PERFORM REQUEST
1320
RETURN DATA AND STATUS TO SPEAKER
1322
EXIT

FIG. 13

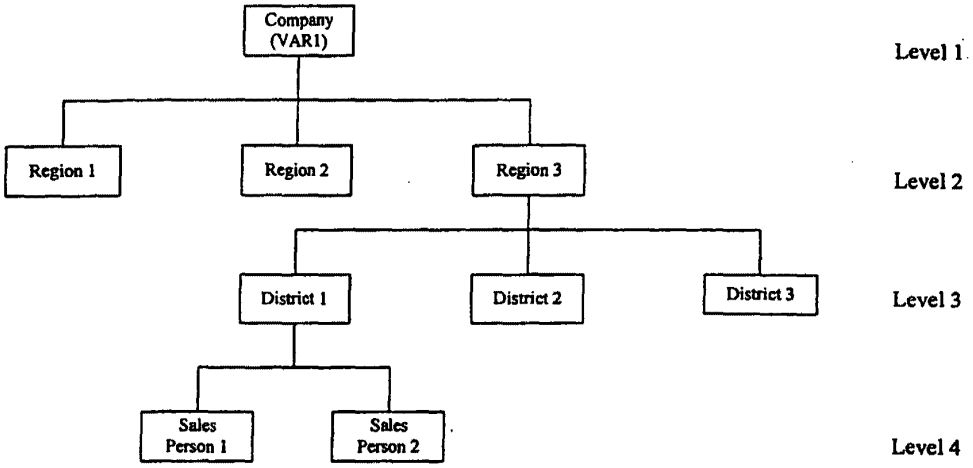

Company (VAR1)
Region 1
Region 2
Region 3
District 1
District 2
District 3
Sales Person 1
Sales Person 2
Level 1
Level 2
Level 3
Level 4

FIG. 14

INTERNATIONAL SEARCH REPORT

Inte. Jonal Application No
PCT/US 99/08061

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 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 6 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)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 98 07088 A (ELECTRONIC DATA SYST CORP) 19 February 1998 (1998-02-19) abstract; figure 1 page 2, line 1 - line 10 page 5, line 10 - line 21 page 10, line 25 - page 11, line 26 claims 1-18 ---	1-36
A	WO 96 42041 A (OPEN MARKET INC) 27 December 1996 (1996-12-27) the whole document ---	1-36
A	US 5 235 642 A (WOBBER EDWARD ET AL) 10 August 1993 (1993-08-10) the whole document --- -/--	1-36

[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
23 August 1999	31/08/1999

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)

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1019

INTERNATIONAL SEARCH REPORT

Inte. .ional Application No
PCT/US 99/08061

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 398 492 A (IBM) 22 November 1990 (1990-11-22) the whole document	1-36

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1020

INTERNATIONAL SEARCH REPORT

Information on patent family members

Inter ..onal Application No
PCT/US 99/08061

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9807088 A	19-02-1998	US 5828833 A	27-10-1998
		AU 3984297 A	06-03-1998
WO 9642041 A	27-12-1996	US 5708780 A	13-01-1998
		US 5812776 A	22-09-1998
		AU 694367 B	16-07-1998
		AU 5936796 A	09-01-1997
		CA 2221506 A	27-12-1996
		EP 0830774 A	25-03-1998
US 5235642 A	10-08-1993	EP 0580350 A	26-01-1994
		JP 6202998 A	22-07-1994
EP 0398492 A	22-11-1990	CA 2016224 A	15-11-1990
		DE 69029759 D	06-03-1997
		DE 69029759 T	17-07-1997
		JP 2060621 C	10-06-1996
		JP 3005868 A	11-01-1991
		JP 7101409 B	01-11-1995
		US 5481720 A	02-01-1996

Form PCT/ISA/210 (patent family annex) (July 1992)

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

(11) EP 0 636 977 B1

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent:
23.05.2001 Bulletin 2001/21

(51) Int Cl.[7]: **G06F 11/00**

(21) Application number: **94305551.7**

(22) Date of filing: **27.07.1994**

(54) **Method and apparatus for detection of computer viruses**

Verfahren und Gerät zur Erkennung von Computerviren

Procédé et appareil de détection de virus d'ordinateurs

(84) Designated Contracting States:
BE DE FR GB IT

(30) Priority: **29.07.1993 US 99368**

(43) Date of publication of application:
01.02.1995 Bulletin 1995/05

(73) Proprietor: **Chambers, David Alan**
Santa Clara, California 95054 (US)

(72) Inventor: **Chambers, David Alan**
Santa Clara, California 95054 (US)

(74) Representative:
O'Connell, David Christopher et al
Haseltine Lake & Co.,
Imperial House,
15-19 Kingsway
London WC2B 6UD (GB)

(56) References cited:
EP-A- 0 510 244

- **IBM TECHNICAL DISCLOSURE BULLETIN, vol. 34, no. 2, 1 July 1991, pages 415-416, XP000211158 "AUTOMATED PROGRAM ANALYSIS FOR COMPUTER VIRUS DETECTION"**
- **IBM TECHNICAL DISCLOSURE BULLETIN, vol. 32, no. 11, 1 April 1990, pages 48-50, XP000097602 "SYSTEM FOR DETECTING UNDESIRED ALTERATION OF SOFTWARE"**

Note: Within nine months from the publication of the mention of the grant of the European patent, any person may give notice to the European Patent Office of opposition to the European patent granted. Notice of opposition shall be filed in a written reasoned statement. It shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

Printed by Jouve, 75001 PARIS (FR)

EP 0 636 977 B1

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1022

Description

BACKGROUND OF THE INVENTION

[0001] The present invention relates generally to a method and apparatus for emulating the execution of a program on a computer system. In particular, the present invention relates to monitoring program behavior to detect and terminate harmful or dangerous behavior in a program. More particularly, the present invention relates to monitoring program behavior to detect computer viruses.

[0002] In recent years, the proliferation of "computer viruses" (generally designed by rogue programmers either maliciously or as "pranks") has become an increasingly significant problem for the owners and users of computer systems. True computer viruses vary, but they share the general characteristic that they comprise executable computer code capable of replicating itself by attachment to and modification of standard computer files. Such files are then considered "infected". On most computer systems, viruses are limited to infecting program applications. When the application is executed, the virus can then replicate and attach copies to further application files. Typically, viruses also engage in other forms of behavior that are considered undesirable, such as re-formatting a hard disk.

[0003] Often grouped with true computer viruses are some other types of malevolent computer programs: worms and trojan horses. Worms do not infect other applications but merely replicate, either in memory or in other storage media. The harmful effect of worms is generally to reduce system performance. Worms are of concern for large multiuser computer systems, but are generally not of concern for personal computers. Trojan horses are programs that masquerade as useful programs or utilities; they generally run only once and have a harmful effect (such as destroying or damaging the computer system data storage). Trojan horses do not replicate, and after being run once by a user, the user is usually alerted to the harmful behavior and will not run the trojan horse again.

[0004] In response to the proliferation of computer viruses, a variety of "antivirus" methodologies and programs have been developed to detect the presence of infected files. These antivirus programs can be generally categorized into groups: behavior interceptors, signature scanners, and checksum monitors.

BEHAVIOR INTERCEPTORS

[0005] The earliest antivirus programs were generally of the behavior interceptor type: they would allow a virus program to execute in memory but would intercept strategic operating system function requests made by the computer virus. Such requests would generally be functions which the virus required to be performed in order to replicate or to destroy its host, i.e., "Write to a file", "Erase a file", "Format a disk" etc. By intercepting these requests, the computer operator/user could be informed that a potentially dangerous function was about to be performed. Control could be halted or continued as necessary. Some antivirus programs actually modify the instructions of the discovered virus program and make them inoperable so as to "kill" them.

[0006] The behavior interceptor method of virus detection has several drawbacks. The first problem is that it relies entirely on user input and decision making when potentially dangerous behavior is detected. This places a great burden on the user, for it is often very difficult to determine whether the flagged behavior is part of the normal operation of the program being executed. For example, disk optimizing programs routinely reformat hard disks to improve the interleave value. In response to a warning message, a user might suspect that their disk optimizer was infected with a virus (when in fact it was not) and halt program execution. Or, worse yet, if the user knows that such behavior is part of the normal operation of a disk optimizer program, they would likely allow the format to continue uninterrupted, which would be disastrous if the program were actually infected.

[0007] A second problem with behavior interceptor antivirus programs is that computer virus technology has advanced to such a state that some computer viruses are able to bypass the interception points used by the antivirus. The virus can then make operating system function requests that are never intercepted by the antivirus, thus avoiding detection.

[0008] A third problem with behavior interceptor antivirus programs is that by allowing the virus to execute, the virus has an opportunity to locate and identify the antivirus program in computer memory. Once the antivirus program is located, the virus can modify the antivirus-- rendering it completely ineffective in exactly the same manner that antivirus programs locate and modify virus programs to render them ineffective.

[0009] A fourth and very significant problem with behavior interceptor antivirus programs is that there are no low level operating system function requests employed by computer viruses that are not also used by any of thousands of non-virus programs. At an instruction by instruction level, or at a function-call by function-call level, a computer virus performs the same operations as legitimate computer programs. In other words, the closer a computer virus is examined, the less distinguishable it becomes from any other computer program.

SIGNATURE SCANNERS

[0010] The next generation of antivirus technology, signature scanners, answered the problem of over-reliance on user interaction as well as the problem of allowing the virus to execute. A signature scanner operates by knowing exactly what a target virus program code looks like ("signature" code) and then scanning for these program codes in any programs requested to be

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1023

executed or otherwise requested to be scanned. As long as the signature codes were sufficiently long enough so as not to be confused with another program's code, then positive identification was virtually guaranteed and the request to execute could be stopped before execution ever began. The primary problem with this technique is that it requires the antivirus developer to have previously collected and analyzed the target viruses, and included the signature codes in the antivirus program. The antivirus program thus relies on an extensive virus signature library, for there are currently several thousand known IBM PC viruses and several new viruses appear each day. Any new viruses appearing after the antivirus program was developed are not included in the library of program codes for which the antivirus can scan. Signature scanning antivirus programs therefore require frequent updates to keep them current with the increasing number of viruses. If the antivirus developer is lax in providing updates, or the user is lax in obtaining and employing available updates, a signature scanning antivirus program can rapidly lose its effectiveness.

CHECKSUM MONITORS

[0011] The last standard technique of virus detection does not look for anything to do with viruses in particular, but concentrates on the host programs which the viruses attack. Every program on a system can be "checksummed" at antivirus installation time. Then, when a virus attaches itself to the unsuspecting host program, the checksum value will (probably) be different and the file infected with the virus can be isolated. The primary problem with this technique is that many programs store varying program information within themselves; this will change the checksum value and thus trigger a false alarm virus detection. Another problem is ensuring the integrity of the checksum information, which is typically attaced to the program file itself or stored in a separate file. Both locations are vulnerable to covert virus modification. Once a virus infects a host, it can then update the stored checksum value to correspond to the newly infected file and then execute undetected.

[0012] IBM Technical Disclosure Bulletin Vol. 34, No. 2, July 1991 pp 415-416 mentions the possibilty of detecting a computer virus by simulating the execution of a file to determine whether such execution would have resulted in the alteration of any executable or other objects in the system or the execution of any self-altering code.

[0013] According to a first aspect of the present invention, there is provided a computer system configured to monitor the execution of a target program, said computer system comprising:

a processing unit having an instruction set;
an instruction emulator for emulating instructions corresponding to the instruction set;
an entry point access controller for controlling access to operating system entry points;
a logger for logging improper access by said instructions to operating system entry points; and
an interrupt tester, wherein said interrupt tester executes a guinea pig file and tests for modification of the guinea pig file to determine if a viral interrupt has been installed.

[0014] According to a second aspect of the present invention, there is provided a computer system configured to monitor the execution of a target program, said computer system comprising;

a processing unit having an instruction set;
an instruction emulator for emulating instructions corresponding to the instruction set;
an entry point access controller for controlling access to operating system entry points;
a logger for logging improper access by said instructions to operating system entry points; and
an interrupt tester, wherein said interrupt tester opens and closes a guinea pig file and tests for modification of the guinea pig file to determine if a viral interrupt has been installed.

[0015] According to a third aspect of the present invention, there is provided, in a computer system, a method for monitoring execution of a target program comprising the steps of;

- emulating instructions corresponding to the instruction set of the target program;
- controlling access to operating system entry points;
- logging improper access by said instructions to operating system entry points; and
- monitoring emulation of the target program to detect a predetermined behaviour indicating presence of a virus, by:

  opening and closing a guinea pig file and testing for modification of the guinea pig file to determine if a viral interrupt has been installed.

[0016] According to a fourth aspect of the present invention, there is provided, in a computer system, a method for monitoring execution of a target program comprising the steps of:

- emulating instructions corresponding to the instruction set of the target program;
- controlling access to operating system entry points;
- logging improper access by said instructions to operating system entry points; and
- monitoring emulation of the target program to detect a predetermined behaviour indicating presence of a virus, by:

  executing a guinea pig file and testing for modification of the guinea pig file to determine if a viral interrupt has been installed.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1024

BRIEF DESCRIPTION OF THE DRAWINGS

[0017] Fig. 1A is a block diagram illustrating the primary components of a computer system executing a target program in a standard manner.

[0018] Fig. 1B is a block diagram illustrating the primary components of a computer system executing a target program according to the present invention.

[0019] Figs. 2A and 2B are diagrams illustrating memory maps of the computer systems of Figs. 1A and 1B, respectively.

[0020] Fig. 3 is a block diagram illustrating the register set emulated by a particular embodiment of the present invention.

[0021] Figs. 4A and 4B are flowcharts illustrating respectively the installation and replication procedures typically employed by computer viruses.

[0022] Fig. 5 is a flowchart illustrating the general emulation process performed by a monitor program according to a particular embodiment of the present invention.

[0023] Fig. 6 is a flowchart illustrating in further detail the memory access control step of the flowchart Fig. 5.

[0024] Fig. 7 is a flowchart illustrating in further detail the procedure access control step of the flowchart of Fig. 5.

[0025] Fig. 8 is a flowchart illustrating in further detail the operating system entry point monitoring step of the flowchart of Fig. 5.

[0026] Fig. 9 is a flowchart illustrating the process performed by a particular embodiment of the present invention to check the behavior of an interrupt handler.

[0027] Fig. 10 is a flowchart illustrating the process performed by a particular embodiment of the present invention to identify viral replication behavior.

[0028] In Fig. 1A a block diagram is shown illustrating the primary components of a computer system executing a target program in a standard manner. The computer system includes a CPU 10, a memory 20, and a disk storage device 30. This is simply an exemplary configuration; the system could of course employ a tape storage device rather than disk storage, and many other variations are possible as well. Operating system 40 typically exists in Read Only Memory, but may also be partially loaded from the disk storage 30 into memory 20. At power up, the CPU begins executing the instructions of operating system 40, which thereafter controls the loading and execution of application programs such as target program 50.

[0029] In this standard configuration, if a user selects target program 50 for execution, operating system 40 would load target program 50 from disk storage 30 into memory 20 and then transfer control to target program 50 by loading the start address of target program 50 into the program counter register, or instruction pointer register, of CPU 10. CPU 10 would then begin executing the instructions of target program 50, as pointed to by the instruction pointer register. Target program 50 will typically include calls to operating system routines, which are identified by a table of pointers, commonly known as interrupt vectors. It is by remapping these interrupt vectors that standard behavior interceptor antivirus programs attempt to maintain control and supervision of target programs. As discussed above, however, many computer viruses are able to circumvent this remapping of the interrupt vectors and are able to use operating system routines without being monitored by the antivirus program.

[0030] In order to prevent this circumvention of monitoring code, a particular embodiment of the present invention is invoked by a user to request that an application program be analyzed for viral behavior. This embodiment takes the form of a monitor program that emulates the execution of the application for a period of time, monitoring its behavior. By emulating the execution of the application program, the application program can be maintained in a controlled environment that cannot be circumvented by a virus.

[0031] The configuration of the monitor program and target application program is illustrated in Fig. 1B. Monitor program 60 loads target program 50 into memory and emulates the execution of the instructions of target program 50, serving as a protective barrier between the application program and the remainder of the computer system. If the application program has not shown any viral behavior at the end of the monitor period, then it is loaded and executed in the standard manner, such as illustrated in Fig. 1A.

[0032] In the secure environment created by the monitor program of Fig. 1B, every aspect of execution can be scrutinized and the operation of the virus can be controlled completely. If the virus were to request a hard disk format operation, a successfully completed status would be returned to it making the virus "believe" that the operation was successful when in fact it was never executed in the first place.

[0033] Figs. 2A and 2B respectively show the general layout in memory 70 for an IBM PC type computer system with a target program loaded directly by PC DOS as in Fig. 1A, and for a target program loaded by an embodiment of the present invention as in Fig. 1B. As shown in Fig. 2A, ROM occupies the upper portion of the memory address space with the remainder of memory being filled up from the bottom: first the operating system 40 in lower memory, followed by device drivers and memory resident programs, then user selected programs such as target program 50. Fig. 2B illustrates memory usage as in Fig. 2A, but additionally with monitor program 60 loaded.

[0034] Fig. 3 illustrates the various CPU registers employed by an 8086 type CPU, the general type of CPU employed by many personal computers, and for which the presently described preferred embodiment is intended. Flags register 300 is a set of bit-wise flags the may be set or cleared during the execution of various types of instructions. These bits can be examined by other in-

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1025

structions to alter program flow or to perform other tasks. Registers 310 are general purpose and are used for a variety of tasks. Index registers 320 are typically used to indirectly reference memory. Stack pointer 330 is used to maintain a data storage stack in memory. Instruction pointer (program counter) 340 points to the location in memory at which the next instruction to be executed resides. Finally, segment registers 350 are used to prepend and additional 4 bits onto other memory addressing registers (16 bits wide), allowing them to access a broader range of memory. Because these registers are intimately involved in the execution of programs, they are all emulated by the monitor program of the preferred embodiment, so as to fully control the execution of a target program.

Viral Code

[0035] Fig. 4A illustrates the installation procedure typically employed by computer viruses. The virus execution begins at block 400 and proceeds to block 410, at which the virus determines if a copy of itself has already been installed in memory. If not, execution precedes to block 420, where the virus the current value of interrupt vector 21h (the operating system entry point on 8086 type computers), and saves this value for later use. Next, at block 430, the virus sets the entry point to point to a procedure within the virus itself, after which at block 440 control is passed to the host program. If at block 410 the virus had determined that a copy had previously been installed, control would pass immediately to block 440.

[0036] Fig. 4B illustrates a typical viral procedure for replication. The beginning of such a procedure would be the replacement entry point stored by the viral code at step 430 of Fig. 4A. When a program later attempts to make an operating system call through int 21, the call would be directed to beginning block 450 of the viral procedure of Fig. 4B. The viral code would then execute, and at block 460 would determine if there was a file name associated with the operating system call. Such operating system calls are typically used by a normal program to open a file or execute another program. If there was a name associated with the operating system call, then at block 470 the viral code would replicate itself by writing its own executable code to the file that was the subject of the operating system call, in some instances after having checked to ensure that this file was not already infected by the virus. After block 470, the viral code would then exit at block 480, passing control to the original interrupt handler, a pointer to which had been saved at block 420 of Fig. 4A. If at block 460 the viral code had determined that there was no filename associated with the operating system call, then execution would have passed directly from block 460 to block 480. In this manner the operating system continues to function normally except for a slight interruption while the viral code executes.

Emulation to Detect Viral Code

[0037] Fig. 5 illustrates the operation of monitor program 60 according to a preferred embodiment of the invention. The monitor program can be executed explicitly by the user with a designated target program, or in alternative embodiments can be executed automatically whenever an operating system call is placed to execute a program. At block 500, the monitor program loads the target program into memory, in exactly the same manner as the operating system would have loaded the target program, but rather than passing execution to the target program immediately, the monitor program retains control for a period of time, to evaluate the target program.

[0038] After the target program is loaded at block 500, at block 510 the monitor program initializes the emulated registers, which correspond to the registers used by CPU 10. These register variables are used by a set of instruction emulation routines that are capable of emulating the instructions of CPU 10. The emulated registers are initialized with the same values that the real registers would have had if the target program had been loaded by the operating system for execution.

[0039] After the emulation registers are initialized, the main emulation loop is entered. At block 520 the instruction pointed to by the emulated program counter register is fetched by the emulation software and the emulated program counter register is incremented by the size of the fetched instruction, so that it points to the next instruction. Control then proceeds to a set of evaluation procedures for the instruction. At block 530, the monitor program determines if the target program is attempting to access memory selected for controlled access. In the preferred embodiment, operating system procedures and data areas the address range of the monitor program are selected for controlled access. Optionally, any memory not belonging to the target program can be selected for controlled access. The memory access process is explained in more detail below with reference to Fig. 6.

[0040] After block 530, at block 540 the monitor program evaluates the instruction for attempted access to a controlled procedure, explained more fully below with reference to Fig. 7, and then also emulates the execution of the instruction. Following block 540 is block 550, at which the monitor program evaluates any possible modifications to the operating system entry points. The processes performed at block 550 are described in more detail below with reference to Figs. 8-10.

[0041] Following the emulation and evaluation blocks 530-550, at block 560 the monitor code determines if the target application has terminated. If so, emulation is terminated at block 570. The determination of step 560 can be according to whether the target program terminates of its own accord, or the determination can be set by a total number of instructions to be emulated or by a fixed period of time for emulation. If the target program

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1026

has not terminated of its own accord at step 560, and if the monitor program has not forcibly terminated it, control returns to block 520, where the next cycle of the emulation loop is begun. The emulation termination at block 570 includes some "cleanup" on the part of the monitor program. This includes displaying to the user a status report of all operating system requests performed by the target program. This step may optionally also include reporting any memory accesses that have been performed outside of the area provided for the target program by the monitor program.

Controlling Access to Memory

[0042] The memory access monitoring process of block 530 is illustrated in further detail in Fig. 6. The described process involves remapping selected parts of memory, which effectively virtualizes those memory areas, making them inaccessible to the target program, and thus protected. In alternative embodiments, access by the target program to these areas of memory is simply denied by the monitor program.

[0043] From the starting point at block 600, the procedure passes to block 610, at which the monitor program determines if the current instruction is one whose function is to access memory. If so, then control passes to block 620, where the monitor program determines if the memory location to be accessed by the current instruction is in an area selected for controlled access. If so, then control passes to block 630, which implements a remapping of the memory address. The monitor program's representation of the instruction is modified to point to the mapping destination, so that the original memory location is protected from the target program.

[0044] In the preferred embodiment, the contents of the original memory location are copied to the mapping destination the first time the location is accessed by the target program. In other embodiments, the contents of the entire memory area selected for controlled access are copied into the mapping destination area when the monitor program first starts. In yet other embodiments, certain areas selected for controlled access can have their mapping destination areas initialized with null or dummy values. For example, it may be desirable that the content of the monitor program be protected and hidden from the target program, so that a virus cannot detect the presence of the monitor program.

[0045] After the remapping of block 630, at block 640 the attempted access to a controlled memory area is logged for later analysis and reporting to the user. After block 640, the memory access control procedure ends at block 650, which returns control to the main process of Fig. 5, at block 540. A negative determination at either of blocks 610 or 620 also results in control passing immediately to block 650.

Controlling Access to Procedures

[0046] In some instances, it is desirable to control access to certain procedures. For instance, operating system procedures, ROM procedures, and interrupt handling procedures can have powerful effects and can be subject to misuse by a virus. For these reasons, it is desirable to control access to them and substitute special purpose procedures in their place, to encapsulate viral code within the emulated environment.

[0047] After the memory access control procedure of block 530 of Fig. 5, control passes to block 540, which is illustrated in further detail in Fig. 7. From beginning block 700 control passes to block 710, at which the monitor program determines if the emulated program counter points to a controlled procedure entry point; a list of such entry points is maintained by the monitor program. If so, then at block 720 the attempted access to a controlled procedure is noted. This can be by displaying a message to the user on the screen, writing to a log file, etc.

[0048] Next, at block 730 the monitor program determines if the instruction is to be directly emulated. This determination is made according to information stored for each controlled procedure entry point; for certain such procedures a special case emulation may be desired rather than directly emulating the instructions of the procedure. If the procedure is not to be directly emulated, then control passes to block 740, where a special case emulation of the entry point instruction is performed. In some instances this special case emulation will entail emulation of the entire controlled procedure at this point.

[0049] If at block 730 it were determined that the controlled procedure was to be directly emulated, or if at block 710 it were determined that the emulated instruction pointer did not indicate a controlled procedure entry point, then control would pass to block 750, where the instruction indicated by the emulated instruction pointer is emulated in the same manner as other instructions. Following the emulation according to either of blocks 740 or 750, control passes to block 760, which returns execution to the main process of Fig. 5

Controlling Access to Operating System Entry Points

[0050] Block 550 is illustrated in further detail by Fig. 8. This control of operating system entry points need not be performed to obtain substantial benefits from the emulation of the target program; however, this process does a higher level of control over the target program and also allows for a more accurate evaluation of viral behavior on the part of the target program.

[0051] From beginning block 800 control passes to block 810, at which the monitor program examines a list of operating system entry points to determine if any have changed as a result of the instruction just emulated. This would indicate that the target program had replaced an

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1027

interrupt handler with a routine of its own. If there is such a change, then it is logged at block 820. At block 820 a flag is also preferable set to indicate that the entry point has changed, so that the change will not be logged redundantly later. In some embodiments, the flag indicates the new value of the entry point, so the monitor program can determine if the entry point gets modified yet again.

[0052] After block 820, at block 830 the emulated instruction pointer, emulated code segment register, and emulated flag register are saved onto the emulated stack. Then the emulated stack pointer is decremented the corresponding 6 bytes, in the same manner as if a hardware interrupt had been received. Next, at block 840, the emulated code segment register and emulated instruction pointer are set to a special purpose monitor program routine to test the interrupt handler just installed by the target program. This interrupt handler testing routine is described below with reference to Fig. 9.

[0053] After block 840, execution passes to block 850, which returns control to the basic process of Fig. 5. This causes the interrupt handler routine of Fig. 9 to be emulated in the same step by step manner as the target program. This maintains the highest degree of encapsulation around the target program, although if detecting viral replication is essentially the only concern, the interrupt handler testing routine of Fig. 9 may alternatively be executed in a more straightforward emulation without many of the execution safeguards described above.

[0054] If at block 810 the monitor program had determined that no operating system entry points had been changed, then control would have passed directly to block 850, and thus returned to the process of Fig. 5 to emulate the remainder of the target program.

Interrupt Handler Testing

[0055] The basic tack of the interrupt handler testing routine is to offer up a guinea pig file for "sacrifice" to a potential viral interrupt handler, and then test the guinea pig file for corruption. This requires that a "clean" guinea pig file already be at hand and also be disposable. This can be easily provided for by several methods, such as by creating the guinea pig file or copying the guinea pig file from a clean library copy at the very start of the monitor program. The guinea pig file should have a known content. It is preferably executable, but without its execution involving writing to other files. The guinea pig file can thus be essentially a null file that does nothing when executed, simply returning immediately.

[0056] As shown in Fig. 9, when the interrupt handler routine is entered at block 900, the first action is to open the guinea pig file, at block 910, after which the guinea pig file is closed at block 920. Next, at block 930 the interrupt handler testing routine examines the guinea pig file to determine if its content has been changed. Such would be the result of a virus having contaminated the interrupt handlers for opening or closing files. If a change is not detected at block 930, then at block 940 the guinea pig file is executed, after which at block 950 the guinea pig file is again examined by the interrupt handler testing routine to determine if its content has been changed by the execution interrupt handler.

[0057] If a positive determination had been made at either of blocks 930 or 950, then execution would pass from the respective block to block 960, at which the unauthorized access to the guinea pig file would be logged. After block 960, and also after a negative determination at block 950, execution passes to block 970, which executes an (emulated) IRET instruction. This is a return from interrupt instruction, which causes the values placed onto the emulated stack at block 830 of Fig. 8 to be restored to the emulated registers. This completes the interrupt handler testing, and returns the emulation to its last point of emulation in the target program.

[0058] For a more refined and definitive degree of analysis, block 960 can also initiate a routine to determine not just if the guinea pig was contaminated, but if it was contaminated in a way so as to contaminate other files; i.e., if it was infected with viral replication behavior. Such a routine is illustrated in Fig. 10. The process of Fig. 10 essentially creates a completely new emulation, with the modified guinea pig file serving as the target program. If this first guinea pig file now passes modification behavior on to a second guinea pig file, then the original target program has been shown to be contaminated with viral code having replicative behavior. To prevent needless additional recursion, the second level of emulation should be identified as such, through use of a flag, etc., so that if block 960 is reached during the second level of emulation, viral behavior is confirmed and the second level of emulation is terminated (rather than beginning another level of testing with yet another guinea pig file).

[0059] This replication detection process is illustrated in Fig. 10. After beginning at block 1000, at block 1010 the complete state of the current emulation is saved, and all operating system entry points, etc., are returned to their values at the beginning of the first emulation. Block 1010 also then includes the step of initiating emulation again, but with the guinea pig file specified as the target program. As noted above, this emulation level should be flagged as a second level emulation.

[0060] Block 1020 indicates the point at which the emulation of the guinea pig file has terminated, after which at block 1040 the first level emulation determines if the emulated guinea pig file had written to a second guinea pig file. This determination is most straightforward if a flag is simply passed from the second level emulation back to the first; it can also be by examining a checksum for the file. If the determination is positive, then at block 1050 the initial target program is confirmed and logged as being virus-contaminated. At block 1060 at the end of the process of Fig. 10, control is passed back to block 970 of Fig. 9, to continue emulation of the initial target program. Alternatively, reaching block 1050 can result

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1028

in the entire emulation being terminated, as the target program has been confirmed as being virus-contaminated.

Alternative Embodiments

[0061] Rather than requiring the user to load the monitor program which then loads the target program, a "zero length loader" TSR version could be installed in a system and every program requested to be executed could be emulated. If no abnormal behavior is found in the first 'n' instructions, the monitor program could pass control to the CPU to allow the target to execute at "full speed" and the end user would not have to be aware of the existence of the monitor program (other than a slight delay during the initial execution).

[0062] Another alternative approach would be where a recursive parser/emulator could effectively evaluate every single instruction of executable code in a program by noting the address of conditional branch instructions, and returning to that branch location, restoring the cpu/memory state of the machine at that instant, and continuing emulation as if the branch had taken the alternate route instead. Emulation continues until all instructions have been evaluated. This would be a time consuming process; however, the information revealed would definitively answer the question of whether the original code was virus infected.

[0063] It is also important to note that, although the described embodiment is oriented towards identifying viral behavior, the disclosed emulation techniques can be constructively employed to emulate program execution in all types of situations where potentially destructive or other predetermined program behavior is a concern.

[0064] It is to be understood that the above description is intended to be illustrative and not restrictive. Many other embodiments will be apparent to those of skill in the art upon reviewing the above description. For instance, the instructions of the emulated application program could be read directly from disk storage rather than being loaded into memory first. The scope of the invention should, therefore, be determined with reference to the appended claims, along with the full scope of equivalents to which such claims are entitled.

Claims

1. A computer system configured to monitor the execution of a target program, said computer system comprising:

   a processing unit having an instruction set;
   an instruction emulator for emulating instructions corresponding to the instruction set;
   an entry point access controller for controlling access to operating system entry points;
   a logger for logging improper access by said instructions to operating system entry points; and
   an interrupt tester, wherein said interrupt tester executes a guinea pig file and tests for modification of the guinea pig file to determine if a viral interrupt has been installed.

2. The computer system of claim 1 wherein said interrupt tester opens and closes a guinea pig file and tests for modification of the guinea pig file to determine if a viral interrupt has been installed.

3. The computer system of claim 1 wherein said interrupt tester executes the guinea pig file as a new target program, thereby creating a second guinea pig file, and tests for modification of the second guinea pig file to determine if a replicative viral interrupt has been installed.

4. The computer system of claim 1 further comprising a procedure access controller for controlling access to procedures during instruction emulation, wherein said logger logs improper access by said instructions to procedures.

5. The computer system of claim 1 further comprising a memory access controller for controlling access by said instructions to memory during instruction emulation wherein said logger logs improper access by said instructions to memory.

6. A computer system configured to monitor the execution of a target program, said computer system comprising;

   a processing unit having an instruction set;
   an instruction emulator for emulating instructions corresponding to the instruction set;
   an entry point access controller for controlling access to operating system entry points;
   a logger for logging improper access by said instructions to operating system entry points; and
   an interrupt tester, wherein said interrupt tester opens and closes a guinea pig file and tests for modification of the guinea pig file to determine if a viral interrupt has been installed.

7. The computer system of claim 6 wherein said interrupt tester executes the guinea pig file as a new target program, thereby creating a second guinea pig file, and tests for modification of the second guinea pig file to determine if a replicative viral interrupt has been installed.

8. The computer system of claim 6 further comprising a procedure access controller for controlling access

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1029

to procedures during instruction emulation, wherein said logger logs improper access by said instructions to procedures.

9. The computer system of claim 6 further comprising a memory access controller for controlling access by said instructions to memory during instruction emulation wherein said logger logs improper access by said instructions to memo

10. Method, in a computer system, for monitoring execution of a target program comprising the steps of;

- emulating instructions corresponding to the instruction set of the target program;
- controlling access to operating system entry points;
- logging improper access by said instructions to operating system entry points; and
- monitoring emulation of the target program to detect a predetermined behaviour indicating presence of a virus, by:

opening and closing a guinea pig file and testing for modification of the guinea pig file to determine if a viral interrupt has been installed.

11. Method, in a computer system, for monitoring execution of a target program comprising the steps of:

- emulating instructions corresponding to the instruction set of the target program;
- controlling access to operating system entry points;
- logging improper access by said instructions to operating system entry points; and
- monitoring emulation of the target program to detect a predetermined behaviour indicating presence of a virus, by:

executing a guinea pig file and testing for modification of the guinea pig file to determine if a viral interrupt has been installed.

Patentansprüche

1. Computersystem, ausgelegt zur Überwachung der Ausführung eines Targetprogramms, beinhaltend:

eine Prozessoreinheit mit einem Befehlssatz;
einen Befehlsemulator zum Emulieren der Befehle entsprechend dem Befehlssatz;
einen Einsprungspunkt-Zugangscontroller zur Regelung des Zugriffs auf die Betriebssystem-Einsprungspunkte;
einen Logger zum Protokollieren eines unberechtigten Zugriffs auf die Betriebssystem-Einsprungspunkte durch Befehle; und
einen Interrupt-Prüfer, wobei der Interrupt-Prüfer eine Versuchdatei ausführt und auf Veränderungen prüft, um herauszufinden, ob ein viraler Interrupt installiert wurde.

2. Computersystem nach Anspruch 1, wobei der Interrupt-Prüfer eine Versuchsdatei öffnet und schließt und auf Veränderungen prüft, um herauszufinden, ob ein viraler Interrupt installiert wurde.

3. Computersystem nach Anspruch 1, wobei der Interrupt-Prüfer eine Versuchsdatei als neues Targetprogramm ausführt, wodurch eine zweite Versuchsdatei eröffnet wird, und die zweite Versuchsdatei auf Veränderungen prüft, um herauszufinden, ob ein replikativer viraler Interrupt installiert wurde.

4. Computersystem nach Anspruch 1, das zudem einen Prozess-Zugangscontroller beinhaltet für die Regelung des Zugriffs auf die Prozesse bei der Befehlsemulation, wobei der Logger einen unberechtigten Zugriff auf die Prozesse durch diese Befehle protokolliert.

5. Computersystem nach Anspruch 1, das zudem beinhaltet einen Speicher-Zugangscontroller zur Regelung des Zugriffs auf die Befehle des Speichers bei der Befehlsemulation. wobei der Logger einen unberechtigen Zugriff auf den Speicher durch diese Befehle protokolliert.

6. Computersystem, ausgelegt zur Überwachung und Ausführung eines Targetprogramms, beinhaltend:

eine Prozessoreinheit mit einem Befehlssatz;
einen Befehlsemulator zum Emulieren der Befehle entsprechend dem Befehlssatz;
einen Einsprungspunkt-Zugangscontroller zur Regelung des Zugriffs auf die Betriebssystem-Einsprungspunkte;
einen Logger zum Protokollieren eines unberechtigten Zugriffs auf die Betriebssystem-Einsprungspunkte durch diese Befehle; und
einen Interrupt-Prüfer, wobei der Interrupt-Prüfer eine Versuchsdatei öffnet und schließt und auf Veränderungen prüft, um herauszufinden, ob ein viraler Interrupt installiert wurde.

7. Computersystem nach Anspruch 6, wobei der Interrupt-Prüfer eine Versuchsdatei als neues Targetprogramm ausführt, wodurch eine zweite Versuchsdatei eröffnet wird, und die zweite Versuchsdatei auf Veränderungen prüft, um herauszufinden, ob ein replikativer viraler Interrupt installiert wurde.

8. Computersystem nach Anspruch 6, das zudem beinhaltet einen Prozess-Zugangscontroller zur Regelung des Zugriffs auf die Prozesse bei der Befehlsernulation, wobei der Logger einen unberechtigen

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1030

Zugriff auf die Prozesse durch die Befehle protokolliert.

9. Computersystem nach Anspruch 6, das zudem beinhaltet einen Speicher-Zugangscontroller zur Regelung des Zugriffs auf die Befehle des Speichers bei der Befehlsemulation, wobei der Logger einen unberechtigen Zugriff auf den Speicher durch die Befehle protokolliert.

10. Verfahren in einem Computersystem zur Überwachung und Ausführung eines Targetprogramms, beinhaltend die Schritte:

Emulieren von Befehlen entsprechend dem Befehlssatz des Targetprogramms;
Überwachen des Zugriffs auf die Betriebssystem-Einsprungspunkte;
Protokollierung eines unberechtigen Zugriffs auf die Betriebssystem-Einsprungspunkte durch die Befehle; und
Überwachen der Emulation des Targetprogramms, um ein vorgegebenes Verhalten, welches auf die Anwesenheit eines Virus hindeutet, zu erfassen, und zwar durch
Öffnen und Schließen einer Versuchsdatei und Prüfen der Versuchsdatei auf Veränderungen, um herauszufinden, ob ein viraler Interrupt installiert wurde.

11. Verfahren zur Überwachung der Ausführung eines Targetprogramms in einem Computersystem, beinhaltend die Schritte;

Emulieren von Befehlen entsprechend dem Befehlssatz des Targetprogramms;
Überwachen des Zugriffs auf die Betriebssystem-Einsprungspunkte;
Protokollieren eines unberechtigen Zugriffs auf die Betriebssystem-Einsprungspunkte durch die Befehle; und
Überwachen der Emulation des Targetprogrammes, um ein vorgegebenes Verhalten, welches auf die Anwesenheit eines Virus hindeutet, zu erfassen, und zwar durch
Ausführen einer Versuchsdatei und Prüfen der Versuchsdatei auf Veränderungen, um herauszufinden, ob ein viraler Interrupt installiert wurde.

Revendications

1. Un système informatique configuré pour contrôler l'exécution d'un programme cible, ledit système informatique comprenant :

une unité de traitement présentant un jeu d'instructions ;

un émulateur d'instructions pour émuler des instructions correspondant au jeu d'instructions ;

un organe de commande d'accès de points d'entrée pour commander l'accès aux points d'entrée du système d'exploitation ;

un consignateur pour consigner un accès incorrect, par lesdites instructions, aux points d'entrée du système d'exploitation ; et

un testeur d'interruption, où ledit testeur d'interruption exécute un fichier cobaye et teste pour modification du fichier cobaye afin de déterminer si une interruption virale s'est installée.

2. Le système informatique de la revendication 1, dans lequel ledit testeur d'interruption ouvre et ferme un fichier cobaye et teste pour modification du fichier cobaye afin de déterminer si une interruption virale s'est installée.

3. Le système informatique de la revendication 1, dans lequel ledit testeur d'interruption exécute le fichier cobaye en tant que nouveau programme cible, en créant ainsi un second fichier cobaye, et teste pour modification du second fichier cobaye afin de déterminer si une interruption virale duplicative s'est installée.

4. Le système informatique de la revendication 1, comprenant en outre un organe de commande d'accès protocole pour commander l'accès à des protocoles pendant l'émulation des instructions, où ledit consignateur consigne un accès incorrecte, par lesdites instructions, aux protocoles.

5. Le système informatique de la revendication 1, comprenant en outre un organe de commande d'accès mémoire pour commander l'accès, par lesdites instructions, à la mémoire pendant l'émulation des instructions où ledit consignateur consigne un accès incorrecte, par lesdites instructions, à la mémoire.

6. Un système informatique configuré pour contrôler l'exécution d'un programme cible, ledit système informatique comprenant :

une unité de traitement présentant un jeu d'instructions ;

un émulateur d'instructions pour émuler des instructions correspondant aux jeux d'instructions ;

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1031

un organe de commande d'accès de points d'entrée pour commander l'accès aux points d'entrée du système d'exploitation ;

un consignateur pour consigner un accès incorrect, par lesdites instructions, aux points d'entrée du système d'exploitation ; et

un testeur d'interruption, où ledit testeur d'interruption ouvre et ferme un fichier cobaye et teste pour modification du fichier cobaye afin de déterminer si une interruption virale s'est installée.

7. Le système informatique de la revendication 6, dans lequel ledit testeur d'interruption exécute le fichier cobaye en tant que nouveau programme cible, en créant ainsi un second fichier cobaye, et teste pour modification du second fichier cobaye afin de déterminer si une interruption virale duplicative s'est installée.

8. Le système informatique de la revendication 6, comprenant en outre un organe de commande d'accès protocole pour commander l'accès à des protocoles pendant l'émulation des instructions, où ledit consignateur consigne un accès incorrecte, par lesdites instructions, aux protocoles.

9. Le système informatique de la revendication 6, comprenant en outre un organe de commande d'accès mémoire pour commander l'accès, par lesdites instructions, à la mémoire pendant l'émulation des instructions où ledit consignateur consigne un accès incorrecte, par lesdites instructions, à la mémoire.

10. Un procédé pour contrôler, dans un système informatique, l'exécution d'un programme cible comprenant les étapes consistant à :

- émuler des instructions correspondant au jeu d'instructions du programme cible ;
- commander l'accès aux points d'entrée du système d'exploitation ;
- consigner un accès incorrecte, par lesdites instructions, aux points d'entrée du système d'exploitation ; et
- contrôler l'émulation du programme cible pour détecter un comportement prédéterminé indiquant la présence d'un virus, en :
  ouvrant et fermant un fichier cobaye et en testant pour modification du fichier cobaye afin de déterminer si une interruption virale s'est installée.

11. Un procédé pour contrôler, dans un système informatique, l'exécution d'un programme cible comprenant les étapes consistant à :

- émuler des instructions correspondant au jeu d'instructions du programme cible ;
- commander l'accès aux points d'entrée du système d'exploitation ;
- consigner un accès incorrecte, par lesdites instructions, aux points d'entrée du système d'exploitation ; et
- contrôler l'émulation du programme cible pour détecter un comportement prédéterminé indiquant la présence d'un virus, en :
  exécutant un fichier cobaye et en testant pour modification du fichier cobaye afin de déterminer si une interruption virale s'est installée.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1032

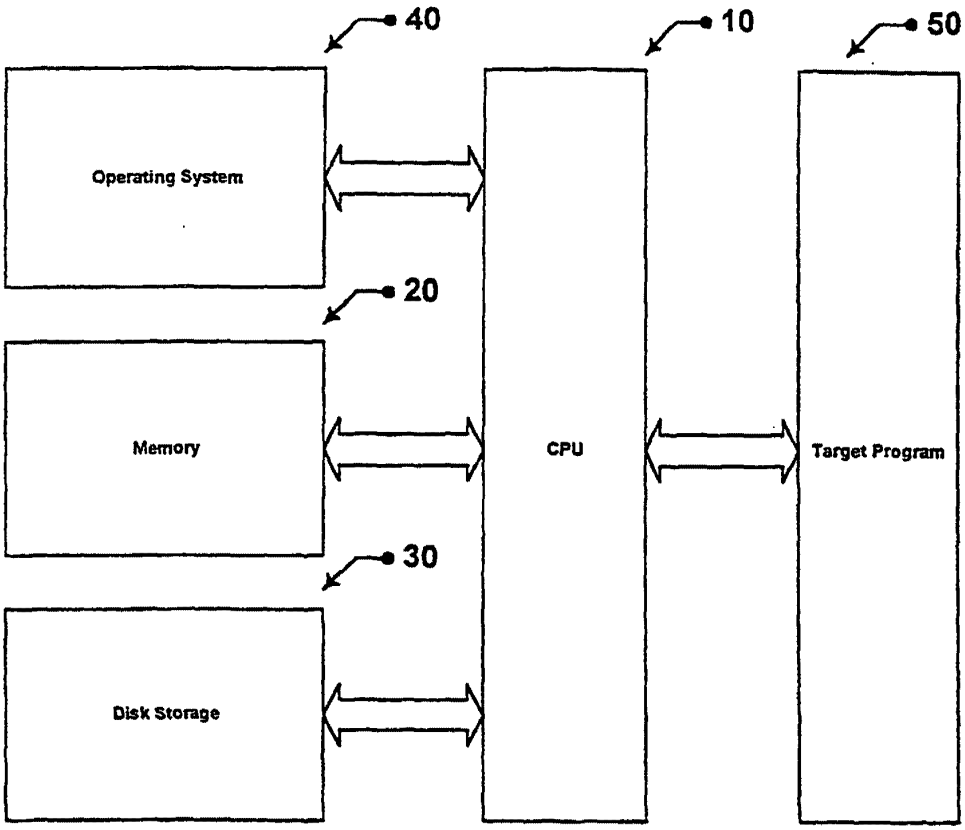


Figure 1A
(Prior Art)

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1033

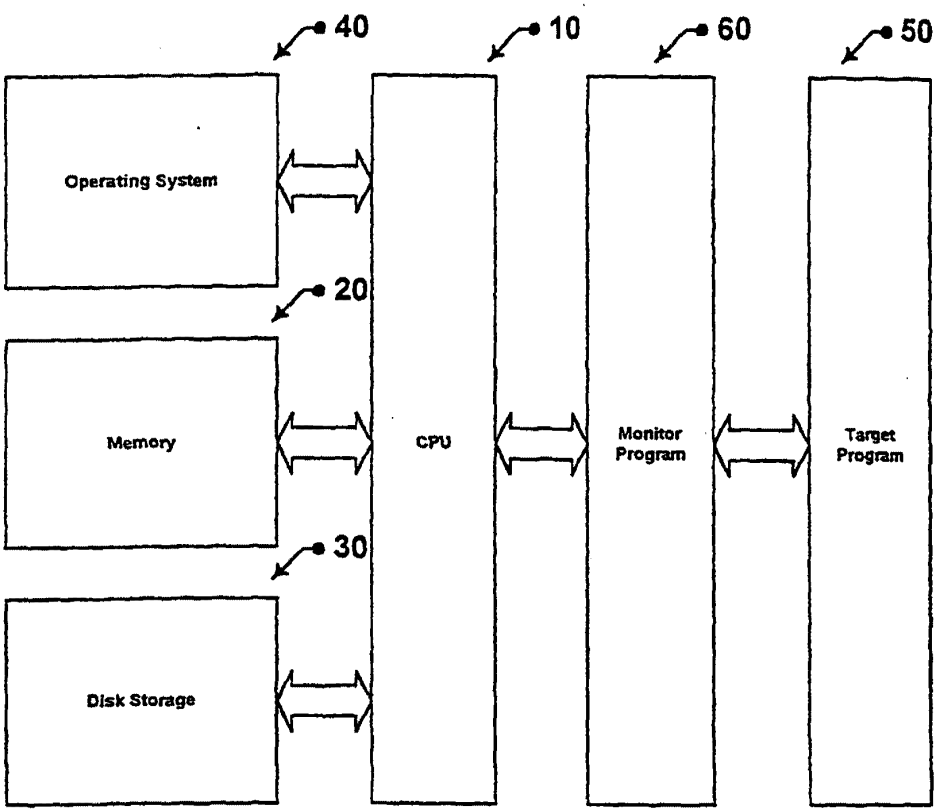


Figure 1B

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1034

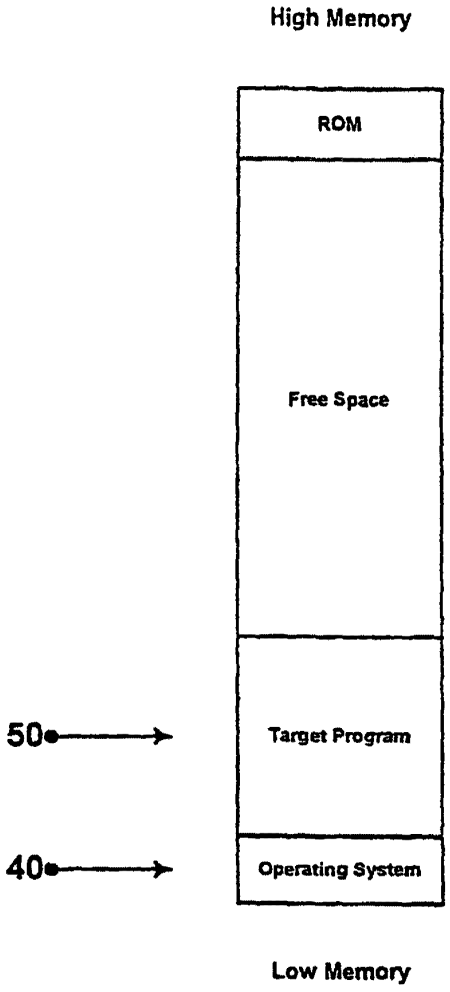


Figure 2A
(Prior Art)

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1035

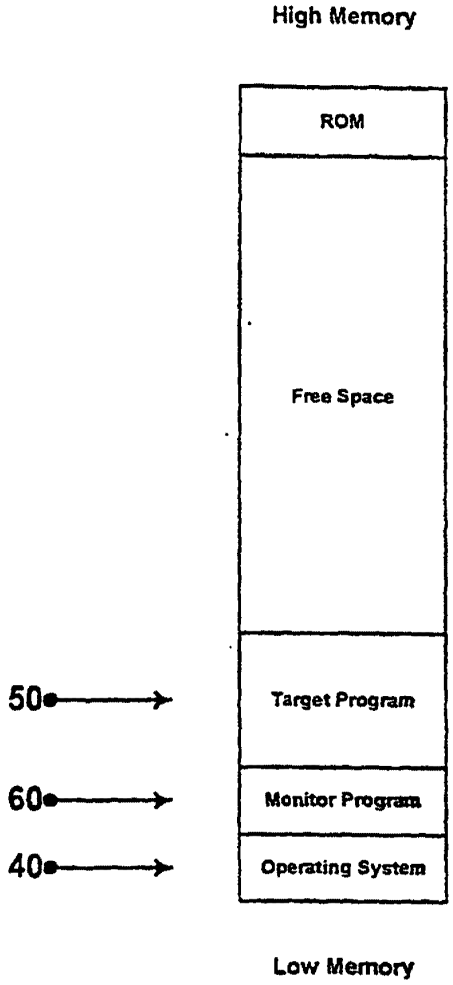


Figure 2B

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1036

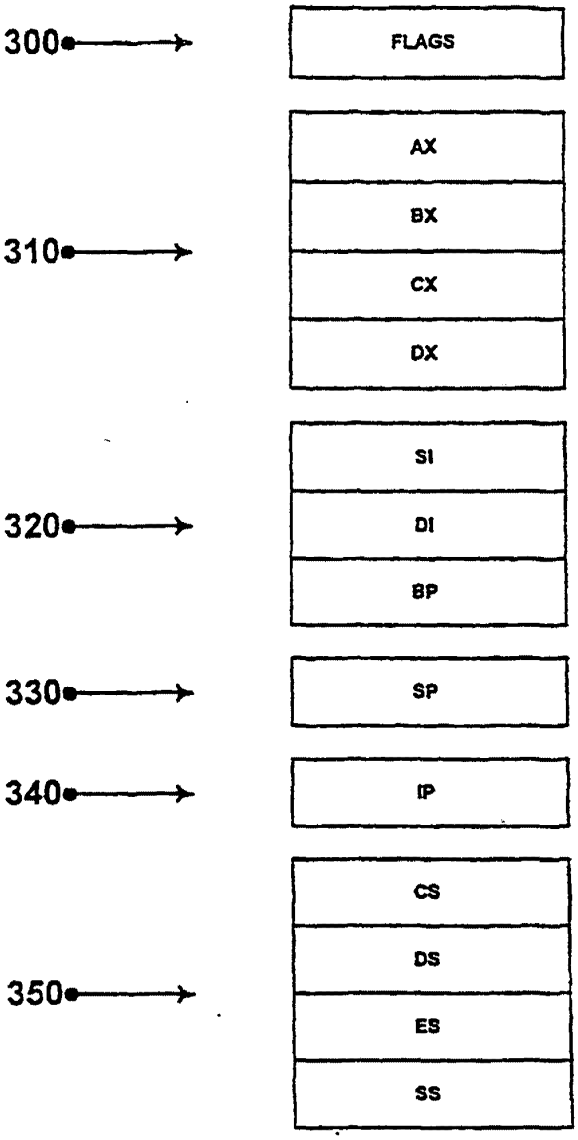


Figure 3
(Prior Art)

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1037

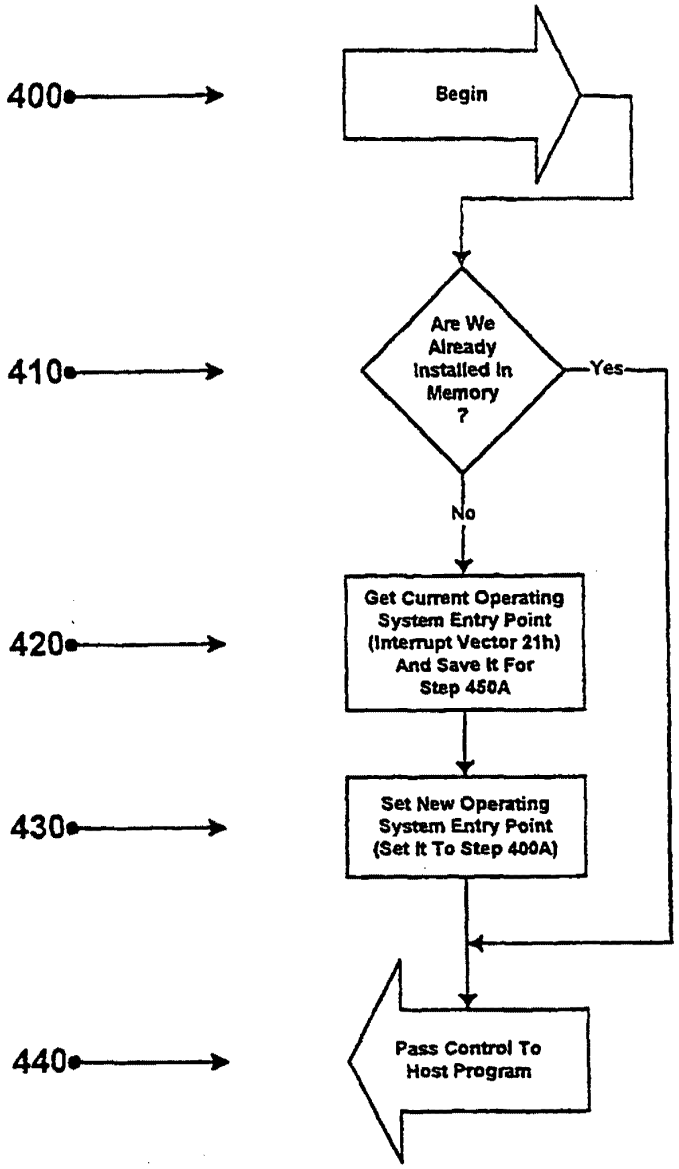


Figure 4A
(Prior Art)

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1038

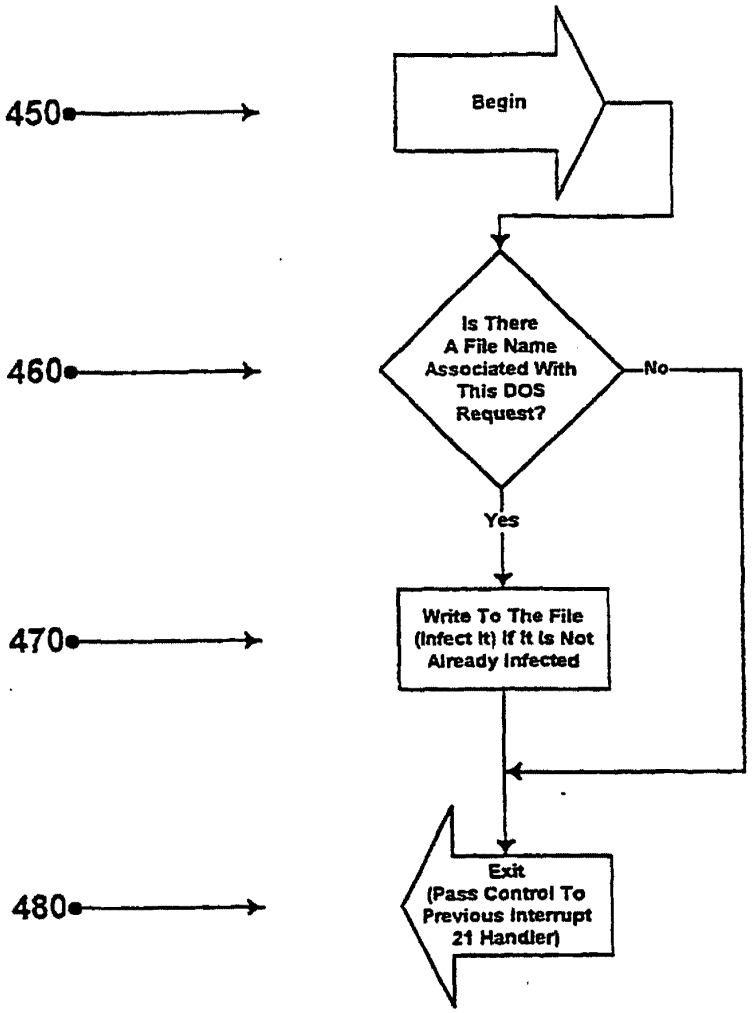


Figure 4B (Prior Art)

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1039

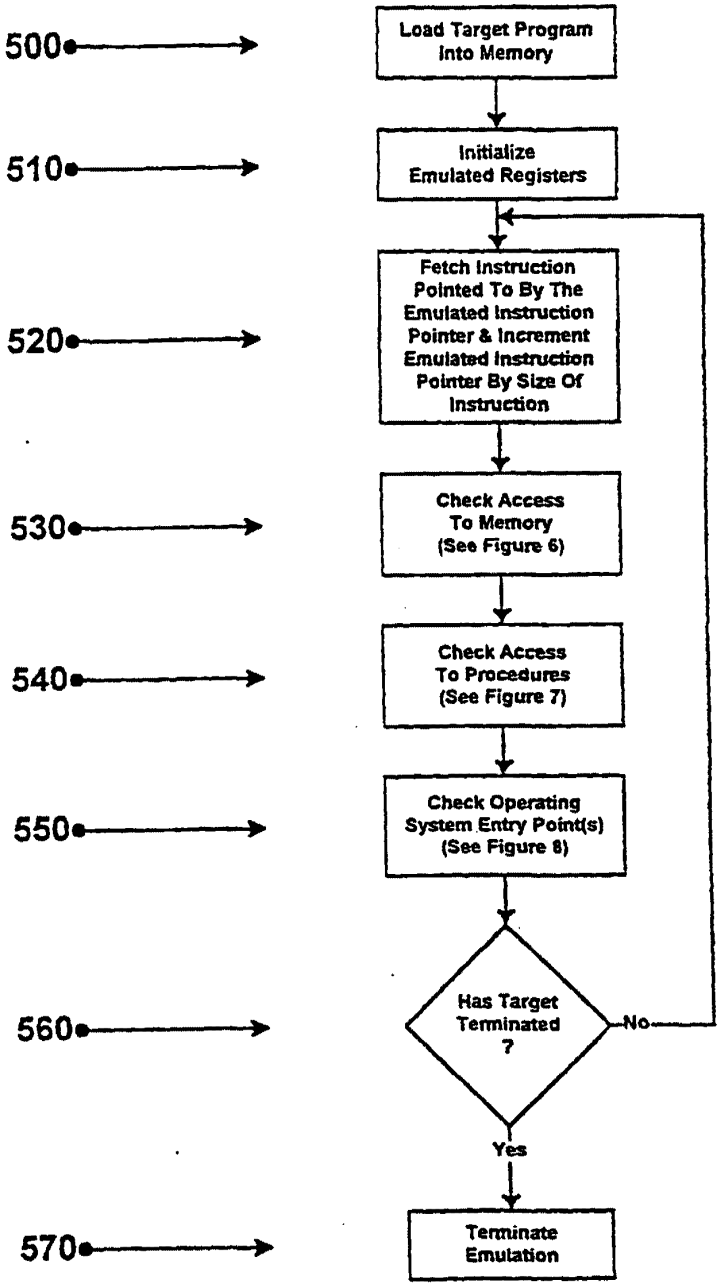


Figure 5

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1040

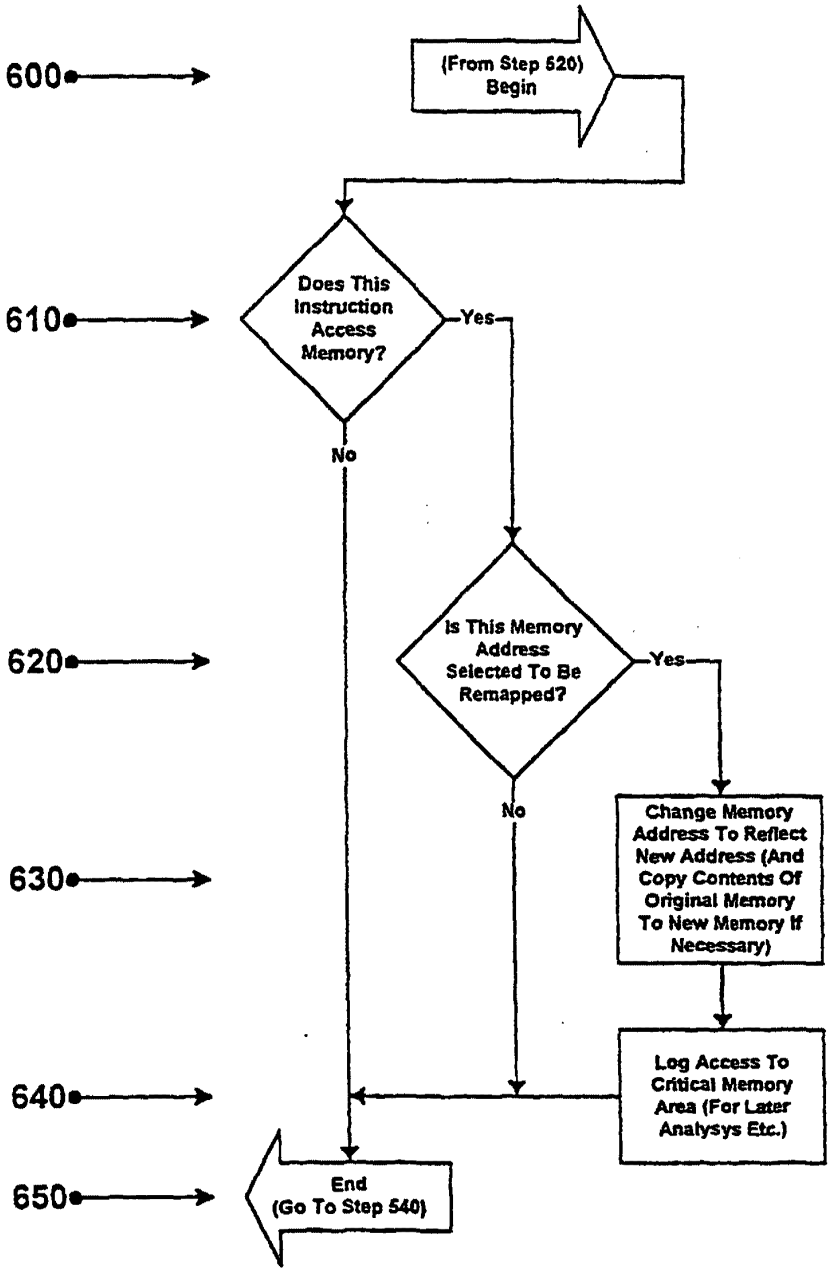


Figure 6

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1041

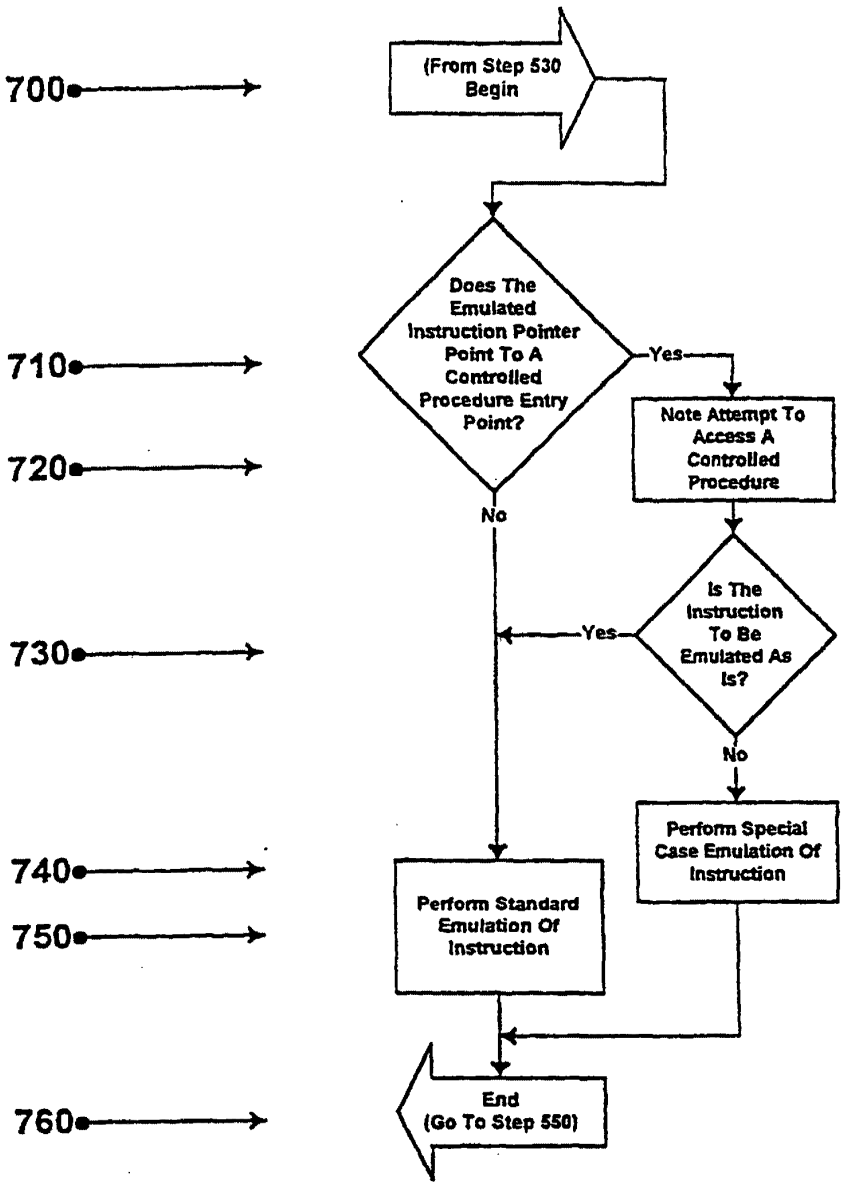


Figure 7

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1042

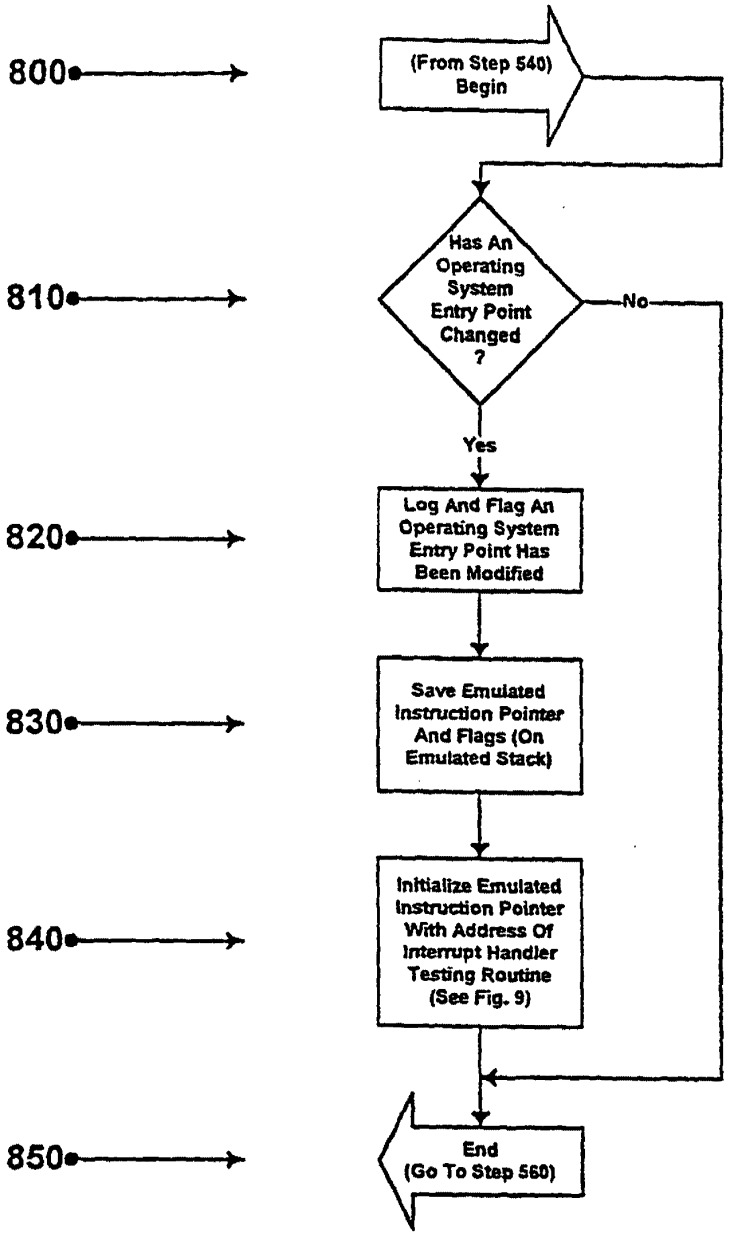


Figure 8

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1043

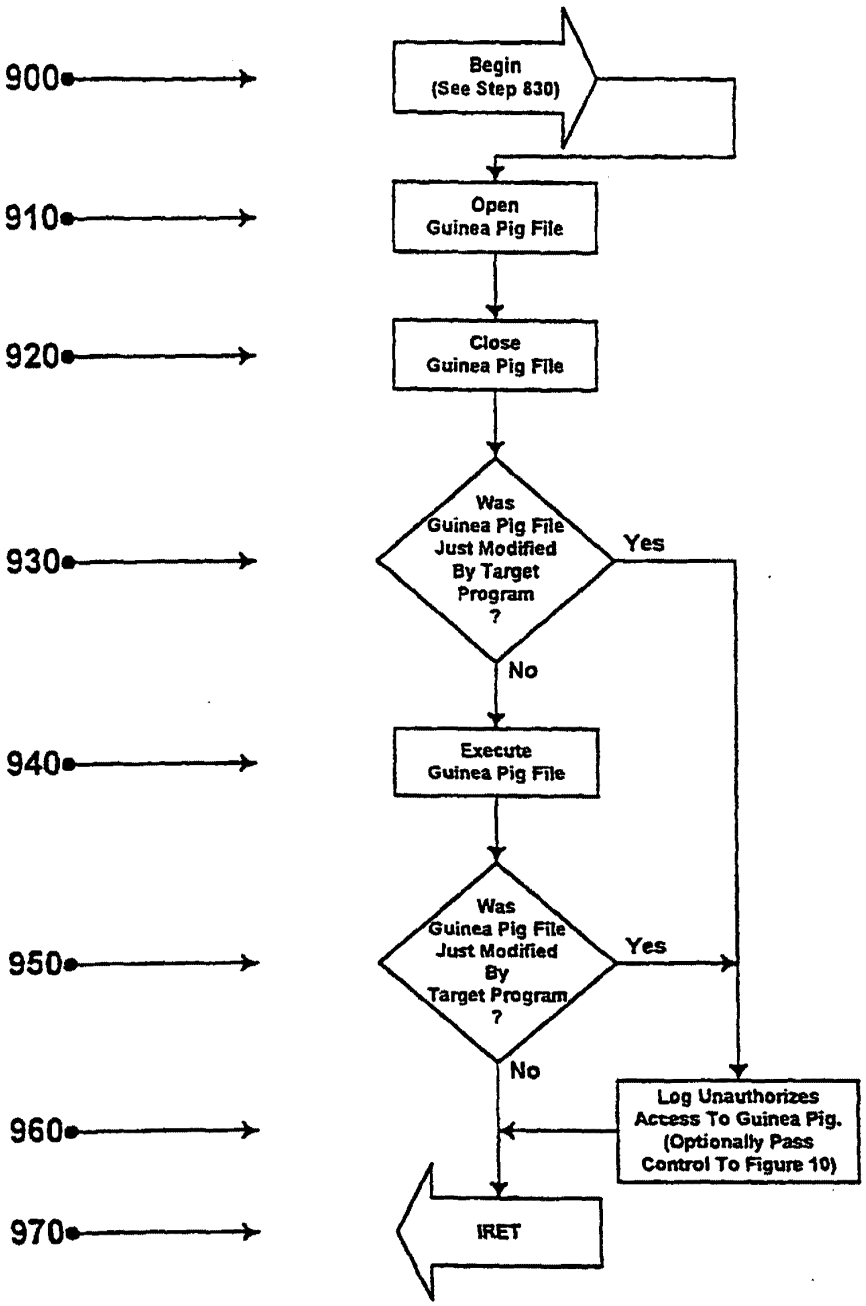


Figure 9

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1044

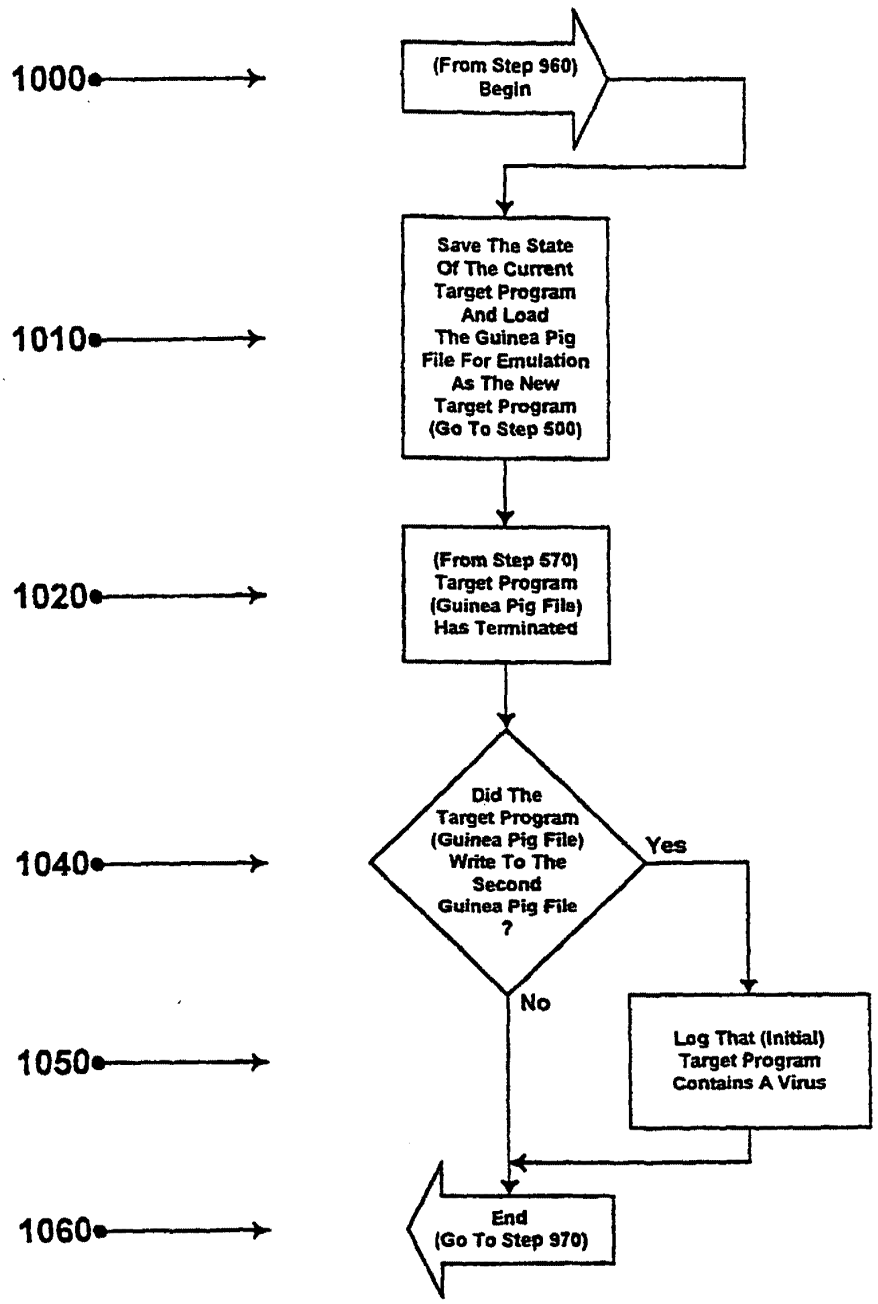


Figure 10

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1045

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 11/00	A1	(11) International Publication Number: **WO 99/15966** (43) International Publication Date: 1 April 1999 (01.04.99)

(21) International Application Number: PCT/US98/17609

(22) International Filing Date: 25 August 1998 (25.08.98)

(30) Priority Data:
08/935,577 23 September 1997 (23.09.97) US

(71) Applicant: SYMANTEC CORPORATION [US/US]; 10201 Torre Avenue, Cupertino, CA 95014 (US).

(72) Inventor: NACHENBERG, Carey, S.; 19585 Shadow Glen Circle, Northridge, CA 91326 (US).

(74) Agents: OKAMOTO, James, K. et al.; Fenwick & West LLP, Two Palo Alto Square, Palo Alto, CA 94306 (US).

(81) Designated States: CA, JP, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: DYNAMIC HEURISTIC METHOD FOR DETECTING COMPUTER VIRUSES

(57) Abstract

A dynamic heuristic method for detecting computer viruses comprising three phases: a decryption phase, an exploration phase, and an evaluation phase. A purpose of the decryption phase is to emulate a sufficient number of instructions to allow an encrypted virus to decrypt its viral body. A purpose of the exploration phase is to emulate at least once all sections of code within a region deemed likely to contain any virus present in the target program. A purpose of the evaluation phase is to analyze any suspicious behavior observed during the decryption and exploration phases to determine whether the target appears to be infected.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1047

DYNAMIC HEURISTIC METHOD FOR DETECTING COMPUTER VIRUSES

I. BACKGROUND

A. Technical Field

This invention relates to computer antivirus software. More particularly, this invention relates to software for detecting unknown computer viruses using emulation and artificial intelligence.

B. Related Art

Computer virus detection technology may be divided into categories such as signature scanning, integrity checking, and non-integrity-based unknown virus detection (also called heuristics). This section discusses these categories of antivirus technology.

Signature scanning antivirus programs work by scanning files for signatures of known viruses. A signature is a sequence of bytes that may be found in a virus program code, yet is unlikely to be found elsewhere. To "extract" a signature, an antivirus researcher must analyze the virus. Once this signature is determined, it is recorded in a database of virus signatures to be used by an antivirus program. The antivirus program scans a target program (executable file, boot record, or possibly document file with a macro) to detect the presence of a virus signature. If a signature is found, then the target program is deemed infected. Otherwise, the target program is considered uninfected.

A signature scanning antivirus program can identify particular virus strains for removal and may have a low "false-positive" rate if properly implemented. However, only viruses whose signatures have already been determined and stored in the signature database may be detected using signature scanning. Moreover, the signature database must be updated frequently to detect the latest viruses.

Integrity checking (called "inoculation" by the commercial Norton Anti-Virus product from Symantec Corp.) is a technique in which "snapshots" or "fingerprints" are taken of programs (executable files, boot records) on the computer under the assumption that all these files are in an uninfected state. These fingerprints are typically taken after the computer has

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1048

been scanned with a virus scanner that reasonably assures the computer is virus-free. These fingerprints are then saved into a database for later integrity-based scans.

During subsequent integrity-based scans of the computer, the antivirus program verifies that each previously fingerprinted program on the computer matches its fingerprint. If a program does not match its fingerprint, then the antivirus program typically uses artificial intelligence to determine if the modification is "virus-like" or merely a valid program update. If the modification appears due to an infection by a virus, the antivirus program typically alerts the user to the modification and gives the user the option to repair the damage, if possible.

Because integrity checking does not scan for virus signatures, it can be used to detect new and (as yet) unknown virus strains. Integrity checking works because viruses must generally make changes to their host program, and these changes can be detected if the database of fingerprints of clean programs is properly created and maintained. However, integrity checking does not work if the computer is not virus-free when the programs are fingerprinted. A virus-infected program that is "inoculated" along with other clean programs would be a safe haven from where the virus can infect other programs. Furthermore, when a change is detected by integrity checking, it is often difficult for the antivirus program to determine if the change was virus-induced or user-induced (e.g., the user may update a program by installing a new version or copying an updated file). If this determination cannot be made by the antivirus program, the user must be called upon to make this determination, and many users are not knowledgeable enough to do so.

Non-integrity-based (also called "heuristic") unknown virus detection is used to detect new and unknown viruses without any integrity information. A heuristic antivirus program examines a target program (executable file, boot record, or possibly document file with a macro) and analyzes its program code to determine if the code appears virus-like. If the target program's code appears virus-like, then the possible infection is reported to the user.

Heuristic virus detection can detect new and unknown viruses that have not yet been analyzed by antivirus researchers since it does not use virus signatures. Because the heuristic technique does not use integrity information, it does not require fingerprints of programs to be taken and saved when the computer is in a known clean state.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1049

Heuristic virus detection can be classified as either static or dynamic. The primary difference between these two detection schemes is that the dynamic method uses CPU emulation while the static method does not.

i. Static Heuristic Detection

In static heuristic virus detection, the antivirus program searches the instructions of a target program for sequences of instructions that perform operations typically used by viruses. Unlike virus signatures, these sequences are not designed to be specific to a single virus. Instead, they are meant to be as general as possible in order to detect the operation of many different viruses.

For example, the following sequence X86 (Intel microprocessor) machine code instructions may be used to open a file:

assembly language	machine code
MOV DX, ????	BA ?? ??
MOV AX, 3D02	B8 02 3D
INT 21	CD 21

where ?? indicates that the byte may vary in different viruses. Similarly, the following sequence of X86 machine code instructions may be used to write to a file:

assembly language	machine code
MOV DX, ????	BA ?? ??
MOV CX, ????	B9 ?? ??
MOV AX, 4000	B8 00 40
INT 21	CD 21

where again ?? indicates that the byte may vary in different viruses.

Static heuristic antivirus programs search for sequences of bytes like those shown above, then makes an assessment of viral infection based on the sequences it found. For example, if the

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1050

static heuristic antivirus program finds a file open operation, followed by file read and write operations, and also finds a character (ASC11) string "VIRUS" in the program, it may report that the file is infected by an unknown virus.

Some (self-decrypting) computer viruses have encrypted viral bodies. Sequences of instructions that exhibit virus-like behavior are not identifiable while they are encrypted. Therefore, some static heuristic detection programs precede the behavior searching phase with a decryption phase which is typically performed using a CPU emulator.

Although static heuristic detection programs can be relatively fast, they may recognize only some of the numerous different ways of performing various virus-like operations. For example, a virus writer may re-order the instructions of the file open sequence above as follows:

assembly language	machine code
MOV AX, 3D02	B8 02 3D
MOV DX, ????	BA ?? ??
INT 21	CD 21.

As a further example, a virus written may more radically change the instructions for a file open as follows:

assembly language	machine code
MOV AX, 3E00	B8 00 3E
SUB AX, 0100	2D 00 01
MOV DX, 1234	BA 34 12
ADD AL, 02	04 02
INT 21	CD 21

Thus, the static heuristic detection program must look for a large number of different ways each virus-like operation may be implemented in order to reliably detect virus-like behavior. A data-base covering large number of possible permutations of these operations may become unmanageable. This problem would be particularly acute if a virus writer wrote a

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1051

"virus generator" program which generated thousands of viruses at a time, permuting the order of its sections of code, but not changing its effective behavior. Such a multitude of viruses would be very difficult to deal with for static heuristic detection programs.

ii. Dynamic Heuristic Detection

In dynamic heuristic virus detection, the antivirus program emulates the target program in a virtual environment and observes the emulated instructions for virus-like operations. As the target program is emulated, its virus-like operations are identified and catalogued. From the catalog of virus-like operations, the dynamic heuristic antivirus program can determine if the target program looks like a virus. Naturally, if the virus has an encrypted viral body, this emulation-based dynamic method can allow the virus to decrypt before observing its virus-like operations (opening files, finding files, etc.).

Dynamic heuristic virus detection can detect many different permutations of a given operation more easily than the static heuristic method. For example, consider the dynamic heuristic detection of a file open operation. Any time an interrupt is called during the emulation, the dynamic heuristic antivirus program checks the values in the registers. These values specify the task that the target program wants the operating system to perform on its behalf. As discussed above regarding static heuristics, a virus infecting the target program may choose to put certain values in the registers in a great variety of ways. However, when the interrupt is finally called, the registers must contain the certain values that correspond to the desired operation. A dynamic heuristic antivirus program is only concerned with the values of the registers at the time of the interrupt call.

While the dynamic heuristic technique is superior in detecting virus-like operations, there are at least three problems to overcome in its implementation. The following is a discussion of these three problems.

First, extensive emulation may be required before the virus-like operations occur. For example, a virus may idle-loop 50,000 times before a file open operation. In that case, a very large number of instructions would have to be emulated before the file open operation is reached. This would greatly slow down the antivirus program.

Second, some viruses activate only when certain arbitrary conditions are met. For example, consider the following pseudo-code of a virus:

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1052

1. Find the first file in the current directory that has a ".com" extension (*.com).
2. If a file was found, go to Step 4.
3. Return control to the host program.
4. If the file is less than 1000 bytes long, go to Step 3.
5. If the file name does not end in "EL", go to Step 3.
6. Open the file.
7. Read the first 3 bytes.
8. Seek to the end of the file.
9. Write virus bytes to the file.
10. etc.

If a dynamic heuristic antivirus program were to emulate a host program infected with such a virus, it would encounter first in the virus Step 1 which instructs to find the first *.com file in the current directory. Here, the antivirus program can simulate the DOS call and indicate to the virus that a mock *.com program was found.

Subsequently, in Step 4, the emulator is instructed to return control from the virus to the host program if the mock *.com program did not have a file size of at least 1000 bytes. How is the antivirus program going to anticipate such an arbitrary condition?

Perhaps the antivirus program will be lucky and the mock *.com program had a file size of at least 1000 bytes. Subsequently, in Step 5, the emulator is instructed to return control from the virus to the host program if the file name does not end in "EL". Once again, if this criterion is not met (e.g., the file name is "FOO.COM," ending in "OO"), the virus will immediately terminate and return control to the host program.

Thus, a virus may be designed to be arbitrarily "picky" in its infection process and if any one criterion (such as the date being the 5th of the month) is not met, the virus will fail to execute its infectious behavior. Consequently, a dynamic heuristic antivirus program will not observe the infectious behavior and will not detect the virus.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1053

Third, while a "direct action" virus (such as the examples discussed above) infects other programs as soon as an infected host program is launched, a "memory resident" virus installs itself as a resident interrupt handler and remains dormant until the appropriate interrupt is called. After installing itself as a resident interrupt handler, the memory resident virus returns control to the host program.

A dynamic heuristic antivirus program begins emulation at the main entry-point of a target program. However, the infectious viral code (the part of the virus that infects other programs) of a memory resident virus is not reached via the main entry-point of its host program. Instead, the infectious viral code is executed only when the interrupt into which the virus is hooked is called, and such a call to the operating system may be made by a different program other than the infected host program.

So, even if the dynamic heuristic antivirus program emulates the infected host program for a very long time, the infectious viral code may not be reached, and thus the suspicious viral operations may go undetected.

II. SUMMARY OF THE INVENTION

The above described problems are overcome by the present invention. The present invention relates to a dynamic heuristic method for detecting computer viruses comprising three phases: a decryption phase, an exploration phase, and an evaluation phase. A purpose of the decryption phase is to emulate a sufficient number of instructions to allow an encrypted virus to decrypt its viral body. A purpose of the exploration phase is to emulate at least once all substantial sections of code within a region deemed likely to contain any virus present in the target program. A purpose of the evaluation phase is to analyze any suspicious behavior observed during the decryption and exploration phases to determine whether the target appears to be infected.

III. BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram illustrating a structure of a computer platform which includes a preferred embodiment of the present invention.

Fig. 2A is a high-level flow diagram of a first and conventional dynamic heuristic method for detecting computer viruses.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1054

Fig. 2B is a high-level flow diagram of a second and preferred dynamic heuristic method for detecting computer viruses according to the present invention.

Fig. 3 is a flow diagram of the decryption phase of the second method according to a preferred embodiment of the invention.

Fig. 4A is a flow diagram of the ninth (analyze/record) procedure of the decryption phase according to a preferred embodiment of this invention.

Fig. 4B is a flow diagram of the tenth (set virus proper) procedure of the decryption phase according to a preferred embodiment of the present invention.

Fig. 5 is a flow diagram of the exploration phase of the second method according to a preferred embodiment of the present invention.

Fig. 6 is a flow diagram of the first (set virtual CS:IP) procedure of the exploration phase according to a preferred embodiment of the present invention.

Fig. 7A is a flow diagram of the fourth (fetch/emulate) procedure of the exploration phase according to a preferred embodiment of the present invention.

Fig. 7B is a flow diagram of the fourth (process instruction) step of the fourth procedure of the exploration phase according to a preferred embodiment of the present invention.

Fig. 8 is a flow diagram of the evaluation phase of the second method according to a preferred embodiment of the present invention.

IV. DESCRIPTION OF THE PREFERRED EMBODIMENTS

Figure 1 is a block diagram illustrating a structure of a computer platform which includes a preferred embodiment of the present invention. The computer platform 100 includes hardware devices 102 and a memory module 104 for storing programs and data. Communications between the memory module 104 and the hardware devices 102 occurs through a communication mechanism 106 (e.g., a bus).

Preferably, the hardware devices 102 include, but are not limited to: a central processing unit (CPU) 108 (e.g., an Intel "x86" or a Motorola PowerPC processor) communicating with: a storage device 110 (e.g., a hard disk drive) and an input/output (I/O) interface 112 (e.g., a small

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1055

computer system interface (SCSI) or a Personal Computer Memory Card International Association (PCMCIA) interface). Communicating with the I/O interface 112 are: a storage drive 114 (e.g., a floppy disk drive), which can read programs and data from a storage medium 116 (e.g., a floppy disk), and a network interface 118 (e.g., an ethernet card or a modem), which communicates with a remote system 120 via a network 122 (e.g., a local area network, a wide area network, an intranet, or the global public Internet). A target software program (or a target data file) suspected of having a computer virus may be stored either on the storage medium 116 (A), the storage device 110 (B), or the remote system 120 (C).

Preferably, the programs and data stored in the memory module 104 include, but are not limited to: an operating system (OS) 150, such as DOS, Windows 3.1, or Windows 95 from Micorsoft (or even Windows NT from Microsoft, or OS/2 from IBM, or Macintosh OS from Apple Computer, or a flavor of UNIX), an antivirus main module 151, a decryption module 152, an exploration module 154, an evaluation module 156, a CPU emulator 158, and virtual memory 160.

A description of the operation of the decryption module 152, the exploration module 154, and the evaluation module 156 begins below in the discussion relating to Fig. 2B.

The CPU emulator 158 executes files in a fully contained virtual environment. The virtual environment includes the virtual memory 160 which is effectively independent of the memory of the host computer system. In this way, the CPU emulator 158 is effectively isolated from the actual hardware devices 102 so that no harm can be done by a virus while a file is being simulated.

Fig. 2A is a high-level flow diagram of a first and conventional dynamic heuristic method 200 for detecting computer viruses. In the first method 200, a single phase 202 emulates the target program and monitors the emulation for virus-like operations. Such a prior art method is described in U.S. Patent No. 5,398,196 which issued on March 14, 1995 to David A. Chambers.

Fig. 2B is a high-level flow diagram of a second and preferred dynamic heuristic method 250 for detecting computer viruses according to the present invention. The second method 250 begins with a decryption phase 252 which involves emulation. A purpose of the decryption phase 252 is to emulate a sufficient number of instructions to allow an encrypted virus to decrypt its viral body. The decryption phase 252 is described below in relation to Fig. 3. When

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1056

the decryption phase 252 is completed, the second method 250 continues with an exploration phase 254 which also involves emulation. A purpose of the exploration phase 254 is to emulate at least once all sections of code within a region likely to contain any virus present, said region being referred to as the virus proper region. In contrast, the emulation phase 202 in Fig. 1 is not so comprehensive as to emulate all sections of code in such a region. The exploration phase 254 is described below in relation to Fig. 5. Following the exploration phase 254, the second method 250 continues with an evaluation phase 256. A purpose of the evaluation phase 256 is to analyze any suspicious operations observed during the decryption and exploration phases 252 and 254 to determine whether or not the target program appears to be infected by a computer virus. The evaluation phase 256 is described below in relation to Fig. 8.

Fig. 3 is a flow diagram of the decryption phase 252 according to a preferred embodiment of the invention. When the antivirus main module 151 is set to scan a target program (A, B, or C in Fig. 1) to determine heuristically whether or not the target program contains virus-like code, the main module 151 begins the decryption phase 252 which includes the following ten procedures 302, 304, 306, 308, 310, 312, 314, 316, 318, and 320..

In a first procedure 302, the decryption module 152 sets the virtual code segment:instruction pointer (CS:IP) of the CPU emulator 158 to the entry-point address of the target program. For example, if the target program is a *.com program, then the virtual IP would be reset to point to the word at 100h in the code segment of the *.com program which is the location of the first executable address of the program. As another example, if the target program is an *.exe program, then the virtual CS would be set to the value of the word at 16h and the virtual IP would be set to the value of the word at 18h in the EXE header as that virtual CS:IP should point to the first executable address of the program.

In a second procedure 304, the decryption module 152 determines whether or not the instruction at the virtual CS:IP was previously modified by another emulated instruction (i.e., whether or not the address of the instruction had been previously written over by a prior emulated instruction). If the instruction at the virtual CS:IP was previously modified, this indicates that a decryption routine of a self-decrypting virus appears to be finished decrypting the viral body and is now trying to execute a decrypted instruction from the viral body. Therefore, if the instruction at the virtual CS:IP was previously modified, then the decryption phase 252 ends and the exploration phase 254 begins.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1057

If the instruction at the virtual CS:IP was not previously modified, then the decryption module 152 performs an third procedure 306 of checking whether or not a first threshold number of instructions have been emulated. The first threshold number of emulated instructions is predetermined such that a typical encrypted viruses would have started to decrypt by the time the first threshold number of instructions have been emulated. In a preferred embodiment, the first threshold number of emulated instructions is three hundred eighty-four (384) instructions. If the first threshold has not yet been reached, then the decryption module 152 continues the emulation by going to a sixth procedure 312, described in more detail below, which fetches the instruction at the CS:IP..

On the other hand, if the first threshold number has been reached, then the decryption module 152 determines in a fourth procedure 308 whether a region of a certain minimum size or larger appears to have been decrypted. Instructions appear to have been decrypted if they have been overwritten. In a preferred embodiment, this certain minimum size is 24 bytes. This certain minimum size of the region is only a fraction (one sixteenth in this case) of the first threshold number of emulated instructions. If no such region appears to have been decrypted, then under the assumption that any virus present is unlikely to be an encrypted virus, the decryption phase 252 ends and the exploration phase 254 begins.

On the other hand, if such a region appears to have been decrypted, then emulation in the decryption phase 252 continues to allow further decryption by fetching the instruction at the virtual CS:IP in the sixth procedure 312 unless a second threshold number of emulated instructions has been reached.

Checking for whether the second threshold has been reached is performed in a fifth procedure 310. In a preferred embodiment, the second threshold is set to about thirty thoursand (30,000) instructions. Once the second threshold number of emulated instructions has been reached, then the decryption phase 252 ends and the exploration phase 254 begins. The purpose of the second threshold is to limit the time taken to allow a virus to decrypt. If the second threshold is reached, the exploration phase 254 begins under the assumption that a significant region of the viral body has been decrypted and exploration of that region is warranted.

In the sixth procedure 312, the decryption module 152 causes the CPU emulator 158 to fetch the instruction stored at the memory address indicated by the virtual CS:IP.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1058

In a seventh procedure 314, the decryption module 152 causes the CPU emulator 158 to increment the virtual CS:IP so that it points to the next instruction in the target program (i.e., to the instruction immediately following the one just fetched).

In a eighth procedure 316, the decryption module 152 causes the CPU emulator 158 to emulate the just-fetched instruction. Because the just-fetched instruction is being emulated by the CPU emulator 158 in a virtual environment, rather than being executed by the actual CPU 108 of the computer system 100, the computer system 100 and its files are protected from any harm from the instruction being emulated.

In a ninth procedure 318, the decryption module 152 analyzes the just-emulated instruction and records any "suspicious" operations indicated by the analysis. This analysis/recording procedure 318 is described in further detail below in relation to Fig. 4A.

In a tenth procedure 320, the decryption module 152 sets the virus proper region. This setting procedure 320 is described in further detail below in relation to Fig. 4B. Following the tenth procedure 320, the decryption module 152 loops back to the second procedure 304.

An advantage of the decryption phase 252 described above is that the substantial majority of self-decrypting viruses are allowed to decrypt much or all of their viral bodies, and this is done without spending too much processing time per file. The decrypted portions of any encrytped viral bodies can then be explored in the subsequent exploration phase 254.

Fig. 4A is a flow diagram which describes a process for the ninth procedure (the analysis/recording procedure) 318 of the decryption phase 252 according to a preferred embodiment of this invention. This analysis/recording procedure 318 begins after the CPU emulator 158 emulates the just-fetched instruction in the eighth procedure 316 of the decryption phase 252.

The ninth procedure 318 begins with a first step 402 in which the decryption module 152 determines whether or not the just-emulated instruction modifies the content stored at a (virtual) memory address within the (virtual) address space of the target program. Such a modification is indicative of decryption taking place.

If such a modification is performed by the just-emulated instruction, then the decryption module 152 performs a second step 404 of recording that modification for future reference (such a reference occurring in the second procedure 304 of the decryption phase 252) before

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1059

performing a third step 406. Otherwise, if such a modification is not performed, then the decryption module 152 goes straight from the first step 402 to the third step 406.

In the third step 406, the decryption module 152 determines whether or not any suspicious operation from a predetermined list of suspicious operations is performed by the just-emulated instruction. The predetermined list of suspicious operations include operations characteristically performed by viruses but not typically performed by "legitimate" uninfected programs.

In this embodiment, the list of suspicious operations includes, but is not limited to, the following suspicious operations:

1. File opens (read-only, read-write, etc.);
2. File reads (sub-categories: 3-4 bytes, between 5 and 8 bytes, 12-14 bytes, 128 or more bytes, etc. These values are chosen because of their particular significance in certain computer viruses.);
3. File writes (sub-categories: 3-4 bytes, between 5 and 8 bytes, 12-14 bytes, 1Ch to 1Eh bytes, 128 or more bytes, etc.);
4. File seeks (to the top of the file, to the end of the file, etc.);
5. File get/set attribute;
6. File get/set date/time;
7. File find-first/find-next;
8. File delete;
9. File create;
10. File rename;
11. Change directory, get current directory, set current directory;
12. File get/set DTA;

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1060

13. Interrupt revectoring (interrupt 13 for disk services, interrupt 21 for DOS services, interrupt 24 for error handling, etc.)

14. Modification of memory allocation (in particular, reads/writes to program segment prefix);

15. Program execution;

16. Program residency self-check calls (typically used by viruses);

17. Read/write to master boot record (MBR) using BIOS disk services (typically used in multipartite and dropper viruses);

18. In-memory modification of CS:100 in *.com programs (typically done in viruses infecting *.com programs).

If it is determined that suspicious operation is being performed by the just-emulated instruction, the suspicious operation is recorded by the decryption module 152 in a fourth step.

In the fifth step 410, the decryption module 152 determines whether or not the just-emulated instruction comprises a branch point. In this specification, a branch point is an instruction which conditionally or unconditionally transfers control to an instruction not sequentially following the branch point instruction. Such a branch point indicates that while one branch of instructions is taken, an alternate branch of instructions is not being taken.

As described above in the background section with respect to the second problem of prior dynamic heuristic antivirus methods, such untaken branches may contain infectious viral code which activates only upon satisfaction of certain arbitrary conditions. For example, consider the following section of assembly language instructions from a hypothetical virus:

```
180   MOV  AX, [TargetHostProgramSize]
185   CMP  AX,1000
188   JLE  210
...
210   HLT
```


IPR2024-00027
CrowdStrike Exhibit 1002 Page 1061

The above section of instructions checks to see if the potential host program that it has located (the target host program) is greater than a minimum size (1000h bytes). If the potential host program is bigger, then it will be considered for infection (i.e. the jump is not taken). Otherwise, the potential host program is not considered for infection and the virus merely halts (i.e. the jump is taken). Branch points such as this one are critical because if the virus does not have its criteria for a host program satisfied, the infectious code may remain in an untaken branch which is not explored by a prior art dynamic heuristic antivirus program.

In light of a need to eventually analyze such untaken branches, if the just-emulated instruction comprises a branch point, then the decryption module 152 in a sixth step 412 stores onto a queue the destination address of the untaken branch (i.e. the address of the first instruction of the untaken branch) and the current state of the CPU emulator 158, before it ends the ninth procedure 318 and begins the tenth procedure 320. The state of the CPU emulator 158 does not refer to the entire virtual CPU/memory state, which is prohibitively large, but rather to a small subset of that state. This subset is sufficient to detect the majority of viruses of current concern. In this embodiment, the state of the CPU emulator 158 includes, but is not limited to, the virtual CS:IP, certain commonly used registers such as the AX, BX, CX, DX registers, and several words at the top of the SS:SP stack.

If only the destination address of the untaken branch, but not the CPU emulator state, were stored, then certain operations would go undetected when that branch of instructions was later explored during the exploration phase 254. For example, consider the following hypothetical section of code from a virus:

```
100   MOV  AX,4200
103   MOV  CX,0000
106   MOV  DX,0000
109   INT  21        INT 21, AX=4200:  SEEK TOP OF FILE
10B   JC   120
10D   MOV  AL,02     INT 21, AX=4202:  SEEK TOP OF FILE
10E   INT  21
```


IPR2024-00027
CrowdStrike Exhibit 1002 Page 1062

...

120 MOV AX,FFFF

123 RET

If the above code were to execute in the emulator and seek instruction at offset 109 were to yield a CARRY result (indicating a seek error), then the jump at offset 10B would be taken to the instruction at offset 120. In that case, the untaken destination address saved upon the queue would be 10D and the state of the CPU emulator 158 saved upon the queue would be CS:10D, AX = 4200, BX = ????, CX=DX=0, the top 8 words of the stack, etc. Later when that untaken branch is explored, the instruction at offset 10D would change AX to 4202, and the instruction at offset 10E would be a seek instruction which is a suspicious operation. However, if the state of the CPU emulator 158 were not saved upon the queue, then the AX register might contain a value such as the FFFF due to an instruction such as the one at offset 120. In that case, when the untaken branch is explored, the instruction at offset 10D would change AX to FF02, and the instruction at offset 10E would not accurately exhibit the operation of the viral code.

Otherwise, if the just-emulated instruction is not a branch point, then the decryption module 152 ends the ninth procedure 318 and begins the tenth procedure 320. The tenth procedure 320 is described in further detail below in relation to Fig. 4B.

Fig. 4B is a flow diagram of the tenth procedure 320 of the decryption phase 252 according to a preferred embodiment of the present invention. This tenth procedure 320 determines a relatively likely location of the virus in the target program.

In a first step 420, the decryption module 152 determines whether or not the virus proper region has already been set (i.e., during a prior performance of either a third step 424 or a sixth step 430 as discussed further below). If the region has already been set, then the tenth procedure 320 ends and the decryption module 152 loops back to the second procedure 304 of the decryption phase 252.

Otherwise, in a second step 422 the decryption module 152 determines whether the target program is an EXE file. If it is an EXE file, then the decryption module 152 in a third step 424 sets the virus proper region to be the region from the start of the beginning code segment (CS) to the end of file for the target program. Following the third step 424, the tenth

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1063

procedure 320 ends and the decryption module 152 loops back to the second procedure 304 of the decryption phase 252.

Otherwise, if the target program is not an EXE file, it is generally a COM file. For COM files, the decryption module 152 in a fourth step 426 determines whether the number of instructions of the target program which have been emulated is greater than n, where n is a predetermined number. In a preferred embodiment, n=10.

If the number of instructions emulated is greater than n, then the decryption module 152 in a fifth step 428 determines whether the current virtual CS:IP is within the beginning memory segment of the target program. If the current virtual CS:IP is outside the beginning segment, then the tenth procedure 320 ends and the decryption module 152 loops back to the second procedure 304 of the decryption phase 252.

Otherwise, if the current virtual CS:IP is within the beginning segment, then in a sixth step 430 the virus proper region is set based upon the current virtual CS:IP. In a preferred embodiment, the region is set to include the 4k (4096) bytes centered about the current virtual CS:IP. Subsequently, the tenth procedure 320 ends and the decryption module 152 loops back to the second procedure 304 of the decryption phase 252.

Fig. 5 is a flow diagram of the exploration phase 254 of the second method according to a preferred embodiment of the present invention. The exploration phase 254 is initiated by the main module 151 after the decryption phase 252, unless the decryption module 152 had returned that the target program appeared to be virus-free. Emulation during the exploration phase 254 is limited to those instructions in the virus proper region. The exploration phase 254 includes the following eight procedures 502, 504, 506, 508, 510, 512, 514, and 516.

In a first procedure 502, the exploration module 154 sets the virtual CS:IP of the CPU emulator 158. The first procedure 502 is described in further detail below in relation to Fig. 6.

In a second procedure 504, the exploration module 154 determines whether the instruction at the virtual CS:IP was already fetched during this exploration phase 504. If the instruction at the virtual CS:IP was not already fetched in this exploration phase 504, then the exploration module 154 determines in a third procedure 506 whether the virtual CS:IP is outside the virus proper region.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1064

If the virtual CS:IP is determined in the third procedure not to be outside the virus proper region, then the exploration module 154 in a fourth procedure 508 causes the CPU emulator 158 to fetch the instruction at the virtual CS:IP and emulate the fetched instruction. The fourth procedure 508 is described in further detail below in relation to Figs. 7A and 7B. After the fourth procedure 508, either the exploration module 154 loops back to the second procedure 504 (if a threshold number of instructions have not yet been emulated), or the exploration phase 254 terminates and the evaluation phase 256 begins (if the threshold number of instruction have been emulated).

On the other hand, if the virtual CS:IP is determined in the third procedure 506 to be outside of the virus proper region, then the exploration module 154 determines in a fifth procedure 510 whether an untaken branch queue is empty. This fifth procedure 510 is also reached if the exploration module determines in the second procedure 504 that the instruction at the virtual CS:IP was already fetched during this exploration phase 254.

If the untaken branch queue is determined in the fifth procedure 510 not to be empty, then the exploration module 154 in a sixth procedure 512 "dequeues" the next untaken branch from the untaken branch queue. Dequeueing the next untaken branch includes setting the virtual CS:IP to the stored destination address of the branch and setting the state of the CPU emulator 158 to the corresponding stored state (both being stored previously during the eighth substep 754 of the fourth step 708 of the fourth procedure 508, see Fig. 7B and the corresponding description below). After the sixth procedure 512, the exploration module 154 loops back to the second procedure 504.

On the other hand, if the untaken branch queue is determined in the fifth procedure 510 to be empty, then the exploration module 154 in a seventh procedure 514 determines whether any unemulated section of code remains within the virus proper region. If such an unemulated section of code remains within the virus proper region, then the exploration module 154 in an eighth procedure 516 resets major virtual CPU registers (to zero) and set the virtual CS:IP to point to the start of the unemulated section of code. After the eighth procedure 516, the exploration module 154 loops back and performs the second procedure 504. On the other hand, if no unemulated section of code remains within the virus proper region, then the exploration phase 254 terminates and the evaluation phase 256 begins. In a preferred embodiment of the present invention, an unemulated section of code will be emulated in the exploration phase 254

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1065

even if the section is only one byte (i.e. one instruction) in size. However, this minimum size may be made larger.

Fig. 6 is a flow diagram of the first procedure 502 of the exploration phase 254 according to a preferred embodiment of the present invention. If no decryption was observed during the decryption phase 252, then the first procedure 502 resets the virtual CS:IP of the CPU emulator 158 to begin the exploration phase 254.

The first procedure 502 begins with a first step 602 in which the exploration module 154 determines whether the decryption was observed during the decryption phase 252. In particular, if the decryption module 152 determined in the sixth procedure 312 of the decryption phase 252 that the just-emulated instruction was previously modified, then the decryption is considered to have been observed.

If decryption was observed in the first step 602, then the virtual CS:IP remains unchanged so that emulation of the target program will continue from the instruction following the last instruction emulated during the decryption phase 252. In this case, after the first step 602, the first procedure 502 of the exploration phase 254 ends and the second procedure 254 begins.

On the other hand, if decryption was not observed in the first step 602, then the exploration module 154 in a second step 604 resets the virtual CS:IP to the entry-point address of the target program.

After the virtual CS:IP is reset in the second step 604, the exploration module 154 in a third step 606 re-establishes the virus proper region. This re-establishing of the virus proper region may be implemented by incorporating a procedure similar to the procedure 320 shown in Fig. 4B into a loop including procedures similar to the procedures 312, 314, and 316. After the virus proper region is re-established in the third step 606, the first procedure 502 of the exploration phase 254 ends and the second procedure 504 begins.

Fig. 7A is a flow diagram which describes a process for the fourth procedure 508 of the exploration phase 254 according to a preferred embodiment of the present invention. The fourth procedure 508 fetches and emulates the instruction at the virtual CS:IP if the virtual CS:IP was determined 506 to be within the virus proper region.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1066

The fourth procedure 508 begins with a first step 702 in which the exploration module 154 updates a bitmap record of instructions fetched from the virus proper region. Such a bitmap record comprises one bit per byte of the virus proper region. If the bit is zero, then the corresponding instruction byte has not yet been fetched. On the other hand, if the bit is one, then the corresponding instruction byte has already been fetched (and emulated). Thus, the bit that corresponds to the instruction byte pointed to by the virtual CS:IP is set to one in this first step 702. Note that this first step 702 need not occur at the very beginning of the fourth procedure 508 and in another embodiment may occur at another point in the fourth procedure 508.

Following the first step 702, the exploration module 154 in a second step 704 causes the CPU emulator 158 to fetch the instruction stored at the virtual memory address pointed to by the virtual CS:IP.

Following the second step 704, the exploration module 154 in a third step 706 causes the CPU emulator 158 to increment the virtual CS:IP so that it points to the next instruction in the target program (i.e., to the instruction immediately following the one just fetched).

Following the third step 706, the exploration module 154 in a fourth step 708 processes the instruction before emulation occurs. This processing 708 exerts control over certain instructions, such as interrupts and memory writes within the virus proper region, and records the presence of these certain instructions for later use if necessary. This fourth step 708 is described in more detail below in relation to Fig. 7B.

Following the fourth step 708, the exploration module 154 in a fifth step 710 determines whether the just-fetched instruction comprises a suspicious operation not already recorded in the fourth step 708. If such suspicious operation was observed, then the exploration module 154 in a sixth step 712 records the suspicious operation. A list of such suspicious operations is predetermined and was generated from an analysis to find viral behaviors or combinations thereof that are very unlikely to be used in non-infected programs. A list of suspicious operations was given above in relation to Fig. 4A. After the sixth step 712, or directly after the fifth step 710 if no suspicious operation was observed, the exploration module 154 in a seventh step 714 causes the CPU emulator 158 to emulate the just-fetched instruction.

After the seventh step 714, the exploration module 154 in an eighth step 716 determines whether a threshold number of instructions have been emulated after the entry-point of the target

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1067

program. In a preferred embodiment, the threshold number is set to the number of bytes in the virus proper region. Since for an Intel X86 microprocessor an instruction is at minimum one byte in length, this threshold number is equal to the maximum number of instructions in the virus proper region. This threshold number ensures that the emulation does not continue much beyond the number of instructions in the virus proper region. If the exploration module 154 determines that the number of instructions emulated since the entry-point has reached the threshold number, then it terminates the exploration phase 254, and the main module 151 causes the evaluation module 156 to begin the evaluation phase 256. On the other hand, if the number of instructions emulated since the entry-point has not yet reached the threshold number, then the exploration module 154 loops back to the second procedure 504 where it determines whether the next instruction was already fetched during the exploration phase 254.

Fig. 7B is a flow diagram which describes a process for the fourth step 708 of the fourth procedure 508 of the exploration phase 254 according to a preferred embodiment of the present invention. This fourth step 708 processes the instruction prior to emulation.

The fourth step 708 begins with a first substep 740 in which the exploration module 154 determines whether or not the just-fetched instruction is a call to an interrupt vector. If it is a call to an interrupt, then the exploration module 154 in a second substep 742 treats the interrupt as a NOP (no operation) instruction but "fakes" completion of the interrupt by changing appropriate registers of the CPU emulator 158, popping appropriate values off of the stack, and so on. In the second substep 742, the exploration module 154 also checks the appropriate register to determine the nature of the actual instruction called via the interrupt, and the actual instruction is recorded if it is deemed to be suspicious.

In a third substep 744, the exploration module 154 determines whether the just-fetched instruction modifies (i.e. writes to) virtual memory within the virus proper region. Such a memory write, for example, may indicates that the virus appears to be relocating its code, or that the virus is storing data within the virus proper region. In this embodiment, such writes are prevented. If the just-fetched instruction does write to memory within the virus proper region, then the exploration module 154 in a fourth substep 746 redirects the memory write to a virtual memory address in a different virtual segment, defining that virtual segment as segment S, and records such a write to segment S.. As discussed below, subsequent data reads from that memory address are redirected to segment S.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1068

The exploration module 154 implements such redirection of subsequent memory reads via fifth and sixth substeps 748 and 750. In the fifth substep 748, the exploration module 154 determines whether the just-fetched instruction initiates a read of data from a location in the virus proper region of memory that the target program previously attempted to modify. If it is determined in the fifth substep 748 that the just-fetched instruction initiates such a memory read of data, then the exploration module 154 in the sixth substep 750 redirects that memory read to segment S. Note, in this embodiment, that fetches from the instruction stream are not to be redirected.

In a seventh substep 752, the exploration module 154 determines whether the just-fetched instruction comprises a branch point. A branch point is defined here as an instruction in a computer program at which a jump occurs or may occur.

If it is determined in the seventh substep 752 that the just-fetched instruction does not comprise a branch point, then the fourth step 708 in Fig. 7A ends and the exploration module 154 goes on to perform the fifth step 710. On the other hand, if it is determined in the seventh substep 752 that the just-fetched instruction is a branch point, then the exploration module 154 in an eight substep 754 stores the current state of the CPU emulator 158 and the destination (also called target) address of the (untaken) branch onto a queue of untaken branches. Subsequently, the fourth step 708 in Fig. 7A ends and the exploration module 154 goes on to perform the fifth step 710.

Fig. 8 is a flow diagram of the evaluation phase 256 which occurs after the exploration phase 254 terminates according to a preferred embodiment of the present invention. The evaluation phase 256 analyzes the suspicious operations observed and recorded during the decryption and exploration phases 252 and 254 to determine whether or not the target program appears to be infected by a computer virus. While this embodiment uses a particular approach to evaluate whether or not the target program is likely to be infected, many other "expert system" approaches may be used in the evaluation phase 256.

In this embodiment, the evaluation phase 256 begins with a first procedure 802 in which the evaluation module 156 determines whether a so-called innocent operation was observed during the decryption or exploration phases 252 or 254. In a preferred embodiment, innocent operations include, but are not limited to:

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1069

1. printing a character to a printer or displaying a character to a monitor or other conspicuous output device;

2. a program termination instruction in the virus proper region that is directly reachable from the entry point;

3. re-vectoring of the divide-by-zero error vector (viruses don't generally need to do division); and

4. calling from within the virus proper region to a function located far (more than four kilobytes in a preferred embodiment) away from the current position in the file (since this would indicate a routine larger than most viruses).

If any of the latter three innocent operations was observed, then the evaluation module 156 determines in a second procedure 804 whether or not specific highly suspicious operations or features were observed. In a preferred embodiment, highly suspicious operations or features include, but are not limited to, the following:

1. comparing a word in any register or memory location with "MZ" (such a comparison may be used to determine the presence of an EXE header since "MZ" is an identifying marker or tag or label of such a header);

2. a file write of the same or a suspiciously similar size in bytes as the distance in bytes from the entry-point of the target program to the end-of-file of the target program (a file write of such a size may be used in the process of appending the virus to the next host program);

3. a seek to the end-of-file to determine file size followed by a file write of a jump instruction whose jump distance is suspiciously similar to the file size (such a jump instruction may be used by an appending virus to bypass the clean original body of the next host program);

4. a file write of a number of bytes similar to the number of bytes in an EXE header where the bytes written include "MZ" (such a file write may be used to write a new header to the next host program);

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1070

5. having an EXE header in the body of an EXE file that is suspiciously similar to the EXE header of the file (such a header may be used to substitute for the original header of the next host program);

6. a command to find executable (e.g., *.com or *.exe) files in a directory (such a command may be used to find the next host program);

7. examination and modification of the (DOS) memory allocation scheme (not uncommon in viruses, but rare in clean programs);

8. repeat move string operations that move approximately the same number of bytes as the distance in bytes from the entry-point to the end-of-file of the target program (string moves of such a total size may be used in the process of appending the virus to the next host program); and

9. reading/writing to the program segment prefix (indicating a modification of the memory allocation scheme in a backhand way).

If no such highly suspicious operation or combination of operations was observed or the innocent operation was printing or displaying a character, then the evaluation module 156 in a third procedure 806 returns a result to the anti-virus main module 151 that indicates that the target program appears to be virus-free.

Otherwise, if any highly suspicious operation was observed 804 or if no innocent operation was observed 802, then the evaluation module 156 in a fourth procedure 808 calculates the extent to which various suspicious operations were observed during the decryption 252 and/or exploration 254 phases. In a preferred embodiment, the fourth procedure 808 involves calculating for each of several suspicious "behaviors" a value based on the number of occurrences of select suspicious operations indicative of the suspicious behavior, where the number of occurrences is weighted depending on the operation's indicativeness regarding the behavior and where the calculated value is normalized so as not to exceed 100%.

In a preferred embodiment, suspicious behaviors include, but are not limited to, the following:

1. encryption;

2. appending bytes onto a COM file;

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1071

3. appending bytes onto an EXE file;

4. opening/closing of a file in standard and non-standard ways;

5. hooking to a memory resident code or routine;

6. critical error handling (re-vectoring a critical error vector);

7. string moves or file writes that are suspiciously similar in size to the target file size;

8. finding a file or finding a first file in a directory; and

9. changing file date/time stamps or other file attributes.

Following the fourth procedure 808, the evaluation module 156 in a fifth procedure 810 determines for each of several types of viruses whether a measure of the suspicious behaviors reaches or surpasses a threshold indicative of infection. In particular, each suspicious behavior is assigned a weight that multiplies its normalized value to arrive at a contribution to the measure. If the measure surpasses a predetermined threshold and every suspicious behavior designated as mandatory for that type of virus is present, then infection by that type of virus is indicated.

In a preferred embodiment, types of viruses include, but are not limited to, the following:

1. memory resident viruses appending to COM files;

2. direct action viruses appending to COM files;

3. memory resident viruses prepending to COM files;

4. direct action viruses prepending to COM files;

5. memory resident viruses appending to EXE files;

6. direct action viruses appending to EXE files;

7. memory resident viruses in general; and

8. direct action viruses in general.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1072

If one (or possibly more) of the thresholds are reached, then in a sixth procedure 812 the evaluation module 156 determines whether any highly suspicious combination of three or more suspicious operations were detected. In a preferred embodiment, these highly suspicious combinations are determined by analyzing various combinations with respect to the universe of known computer viruses and selecting as highly suspicious those combinations which appear in several (e.g., five) percent or more of the known viruses. Other ways of selecting the highly suspicious combinations are possible. When a highly suspicious combination generates a false positive by incorrectly identifying a clean program as infected, then that highly suspicious combination may be removed from the list of highly suspicious combinations. Such a list should be quite robust since more than one highly suspicious combination may be expected to detect a typical virus.

If a highly suspicious combination is detected 814, then the evaluation module 156 performs the sixth procedure 812 of returning to the anti-virus main module 151 the type of infection apparently present in the file. In particular, a label corresponding to the combination(s) detected is returned.

Otherwise, if no highly suspicious combination is detected, then the evaluation module 156 performs the third procedure 806 of returning to the anti-virus main module 151 the result that the file appears to be virus-free.

The above description is included to illustrate the operation of the preferred embodiments and is not meant to limit the scope of the invention. From the above description, many variations will be apparent to one skilled in the art that would yet be encompassed by the spirit and scope of the invention. First, groups of blocks in Figs. 4A and 7B may be reordered without changing the functionality of the embodiment. (For example, steps 402 and 404 may be moved to be after steps 406 and 408. Or, steps 740 and 742 in Fig. 7B may be moved to be after steps 748 and 750. And so on.) Second, the determination of when to end the decryption phase 252 may be done in alternate ways different from the way using two thresholds which is shown in Fig. 3. Third, the determination of the virus proper region may be done in many alternate ways different from the specific way which is shown in Fig. 4B. Fourth, the evaluation of whether or not the file appears infected may be done in alternate ways different from the vector approach of the evaluation phase 256 illustrated in Fig. 8. Fifth, different criteria, or combinations of criteria, for suspicious operations and behaviors may be used in the evaluation phase 256 that vary from the criteria given in the specification. These are only some of the

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1073

variations that are rightly encompassed by the present invention. The scope of the invention is to be limited only by the following claims. What is claimed is:

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1074

CLAIMS

1. A method for detecting undesirable code in a digital file comprising:

determining a region of the digital file in which the undesirable code is likely to be present; and

emulating at least once all substantial sections of code within the region in order to fully explore the region during an exploration phase.

2. The method of claim 1, wherein the region includes an end portion of the digital file.

3. The method of claim 1, wherein the region includes a beginning portion of the digital file.

4. The method of claim 1, further comprising:

pointing to an instruction to be emulated from a section of code;

determining whether during the exploration phase the instruction to be emulated has already been emulated; and

if the instruction to be emulated has already been emulated during the exploration phase, then not emulating the instruction to be emulated.

5. The method of claim 4, wherein the emulating further comprises:

determining whether the instruction to be emulated is outside of the region; and

if the instruction to be emulated is outside the region, then not emulating the instruction to be emulated.

6. The method of claim 5, further comprising:

determining whether any untaken branch destination remains; and

if any untaken branch destination remains, then pointing to a next untaken branch destination as the instruction to be emulated and setting a state of a CPU emulator to a stored state which corresponds to the next untaken branch destination.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1075

7. The method of claim 6, further comprising:

if no untaken branch destinations remain, then determining whether any unemulated section of code remains within the region; and

if any unemulated section of code remains within the region, then pointing to a beginning instruction of an unemulated section of code as the instruction to be emulated and resetting the state of the CPU emulator.

8. The method of claim 5, wherein emulating the instruction to be emulated comprises:

determining whether the instruction to be emulated performs a suspicious operation; and

recording the suspicious operation if the instruction to be emulated performs the suspicious operation.

9. The method of claim 5, wherein emulating the instruction to be emulated comprises:

determining whether the instruction to be emulated comprises an interrupt; and

if the instruction to be emulated comprises an interrupt, then faking completion of the interrupt but not actually performing the interrupt.

10. The method of claim 5, wherein emulating the instruction to be emulated comprises:

determining whether the instruction to be emulated comprises a memory write to a location within the region;

redirecting the memory write to a different segment S if the instruction to be emulated comprises a memory write to a location within the region.

11. The method of claim 10, wherein emulating the instruction to be emulated further comprises:

determining whether the instruction to be emulated comprises initiating a memory read of data previously written by redirection to segment S; and

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1076

redirecting the memory read of data to the segment S if the instruction to be emulated comprises the memory read of data previously written by redirection to the segment S.

12. The method of claim 5, wherein emulating the instruction to be emulated comprises:

determining whether the instruction to be emulated comprises a branch point;

storing an untaken destination address of the branch point and a CPU emulator state if the instruction to be emulated comprises a branch point.

13. A method for detecting a computer virus in a target program comprising:

a first emulation of instructions of the target program to allow the computer virus to decrypt a viral body; and

a second emulation of instructions of a region of the target program which is likely to contain the viral body.

14. The method of claim 13, wherein the first emulation begins at an entry-point instruction of the target program.

15. The method of claim 13, wherein the second emulation begins at an instruction immediately following a last instruction emulated during the first emulation.

16. The method of claim 13, wherein during the first emulation the computer virus does not decrypt the viral body because the viral body is not encrypted; and wherein the second emulation begins at an entry-point instruction of the target program.

17. The method of claim 13, wherein whether the viral body has decrypted is determined by identifying a contiguous section of bytes in memory which were overwritten during the first emulation.

18. The method of claim 13, further comprising an evaluation of suspicious operations observed during the first and second emulation.

19. The method of claim 18, wherein the evaluation of suspicious operations results in a determination of whether the target program appears to be infected.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1077

20. A method for detecting undesirable code in a program comprising: a decryption phase, an exploration phase; and an evaluation phase.

21. A method for detecting computer viruses in a target program comprising:

emulation of instructions of the target program;

evaluation of operations observed during the emulation; and

stopping the evaluation if an operation was observed which is rarely found in the computer viruses.

22. The method of claim 21, wherein the operation comprises printing or displaying a character.

23. The method of claim 21, wherein the operation comprises a program termination instruction within a particular region of the program.

24. The method of claim 21, wherein the operation comprises a re-vectoring of an error vector.

25. The method of claim 21, wherein the operation comprises calling from within a particular region of the program a function far away from a present location.

26. The method of claim 21, wherein the stopping is avoided if a highly suspicious operation is observed.

27. The method of claim 26, wherein the highly suspicious operation comprises comparing contents of a register or memory location with an identifier of a file header.

28. The method of claim 26, wherein the highly suspicious operation comprises a file write of similar size in bytes as a distance in bytes from an entry-point of the target program to an end-of-file of the target program.

29. The method of claim 26, wherein the highly suspicious operation comprises a seek to an end-of-file followed by a file write of a jump instruction.

30. The method of claim 29, wherein the seek determines a size in bytes and the jump instruction has a jump distance of a similar size in bytes.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1078

31. The method of claim 26, wherein the highly suspicious operation comprises a file write of a similar size as a size of a header, and the file write includes a tag that is characteristic of the header.

32. The method of claim 26, wherein the highly suspicious operation comprises having a header of an executable file within a body of the target program.

33. The method of claim 26, wherein the highly suspicious behavior comprises finding an executable file in a directory.

34. The method of claim 26, wherein the highly suspicious behavior comprises examination or modification of the memory allocation scheme.

35. The method of claim 26, wherein the highly suspicious behavior comprises one or more move string operations that move approximately a same number of bytes as a distance in bytes from an entry-point of the target program to an end-of-file of the target program.

36. A method for detecting undesirable code, such as a computer virus, trojan horse, or worm, in a computer file comprising:

determining a region of the computer file which is likely to contain the undesirable computer code; and

emulating substantially all code in the region.

37. A method for detecting undesirable code in digital code comprising:

determining a presence of a first feature in the digital code;

determining a presence of a second feature in the digital code;

determining a presence of a third feature in the digital code;

determining a logical value based upon a logical combination of the presence of the first, second, and third features; and

indicating the presence of the undesirable code if the logical value is true.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1079

38. A method for detecting undesirable code in a target program including detecting a presence of an operation comprising comparing contents of a register or memory location with an identifier of a file header.

39. A method for detecting undesirable code in a target program including detecting a presence of an operation comprising a file write of a similar size in bytes as a distance in bytes from an entry-point of the target program to an end-of-file of the target program.

40. A method for detecting undesirable code in a target program including detecting a presence of an operation comprising a seek to an end-of-file followed by a file write of a jump instruction.

41. The method of claim 40, wherein the seek determines a size in bytes and the jump instruction has a jump distance of a similar size in bytes.

42. A method for detecting undesirable code in a target program including detecting a presence of an operation comprising a file write of a similar size as a size of a header, and the file write includes a tag that is characteristic of the header.

43. A method for detecting undesirable code in a target program including detecting a presence of an operation comprising having a header of an executable file within a body of the target program.

44. A method for detecting undesirable code in a target program including detecting a presence of an operation comprising finding an executable file in a directory.

45. A method for detecting undesirable code in a target program including detecting a presence of an operation comprising examination or modification of the memory allocation scheme.

46. A method for detecting undesirable code in a target program including detecting a presence of an operation comprising one or more move string operations that move approximately a same number of bytes as a distance in bytes from an entry-point of the target program to an end-of-file of the target program.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1080

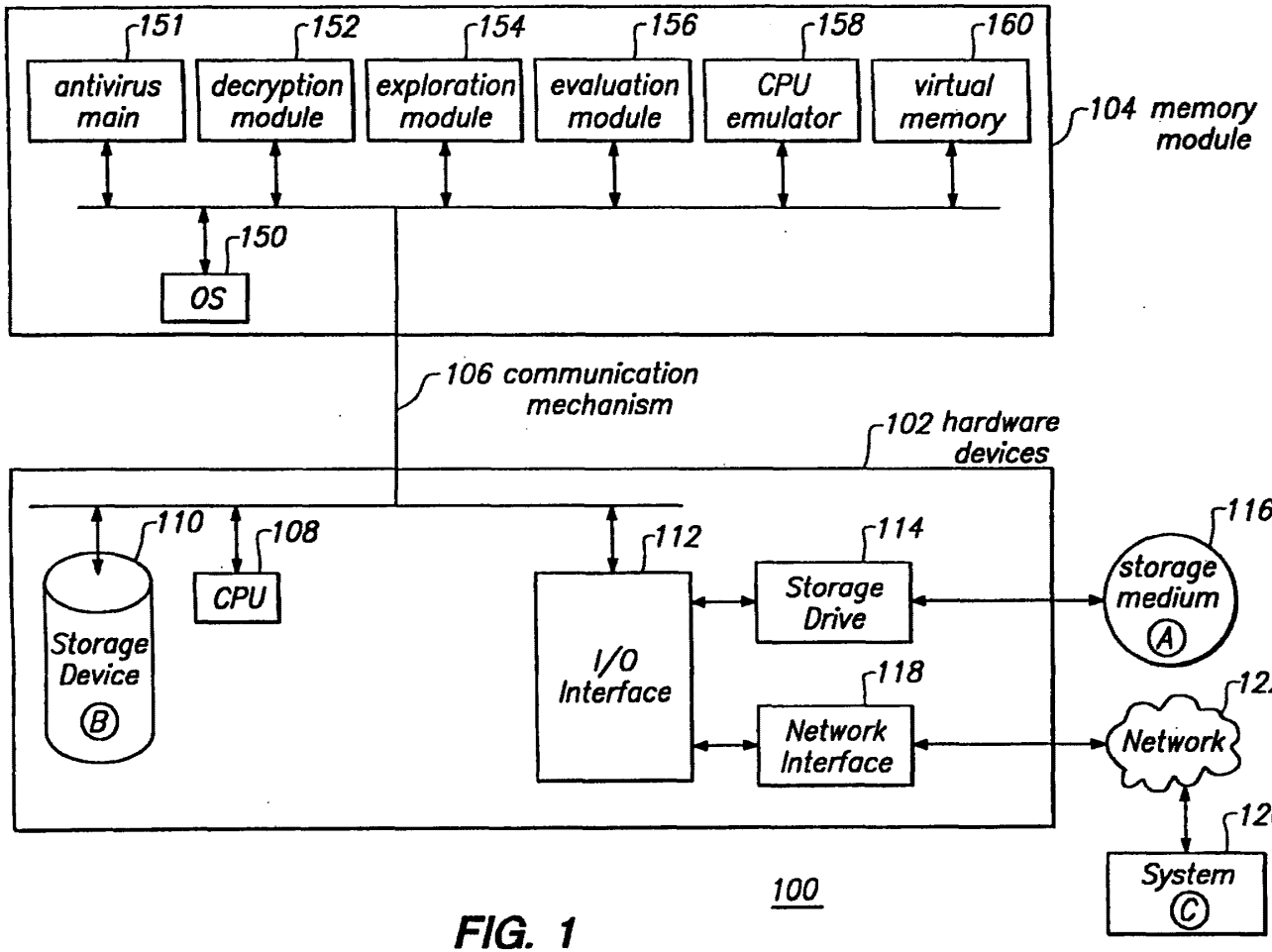

151
152
154
156
158
160
antivirus main
decryption module
exploration module
evaluation module
CPU emulator
virtual memory
104 memory module
150
OS
106 communication mechanism
102 hardware devices
110
108
112
114
116
Storage Device
B
CPU
I/O Interface
Storage Drive
storage medium
A
118
122
Network Interface
Network
120
System
C
100

FIG. 1

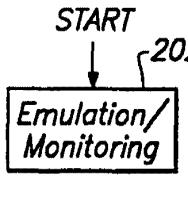


FIG. 2A (Prior Art)

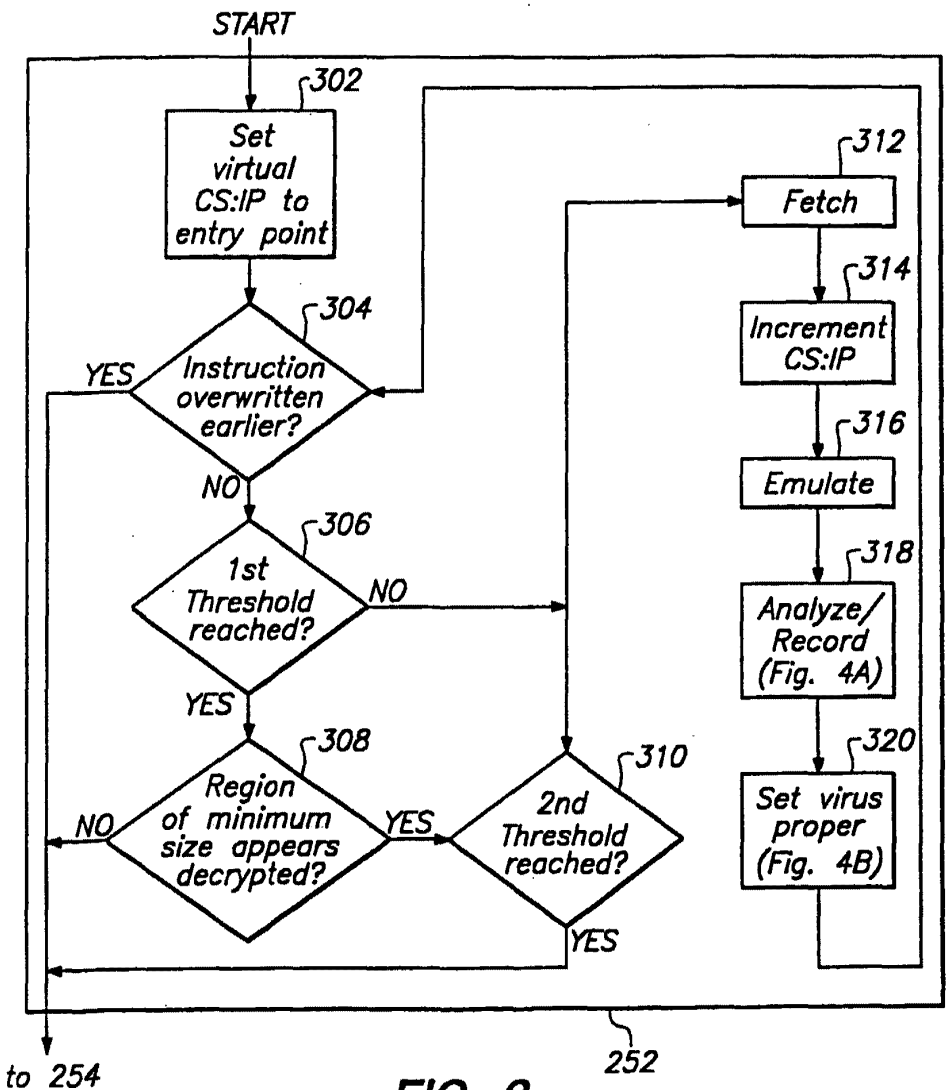


FIG. 3

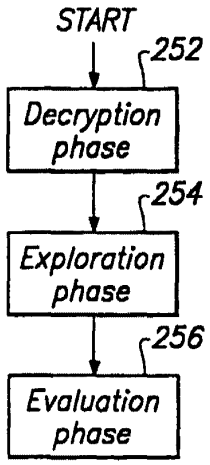

START
252
Decryption phase
254
Exploration phase
256
Evaluation phase
250

FIG. 2B

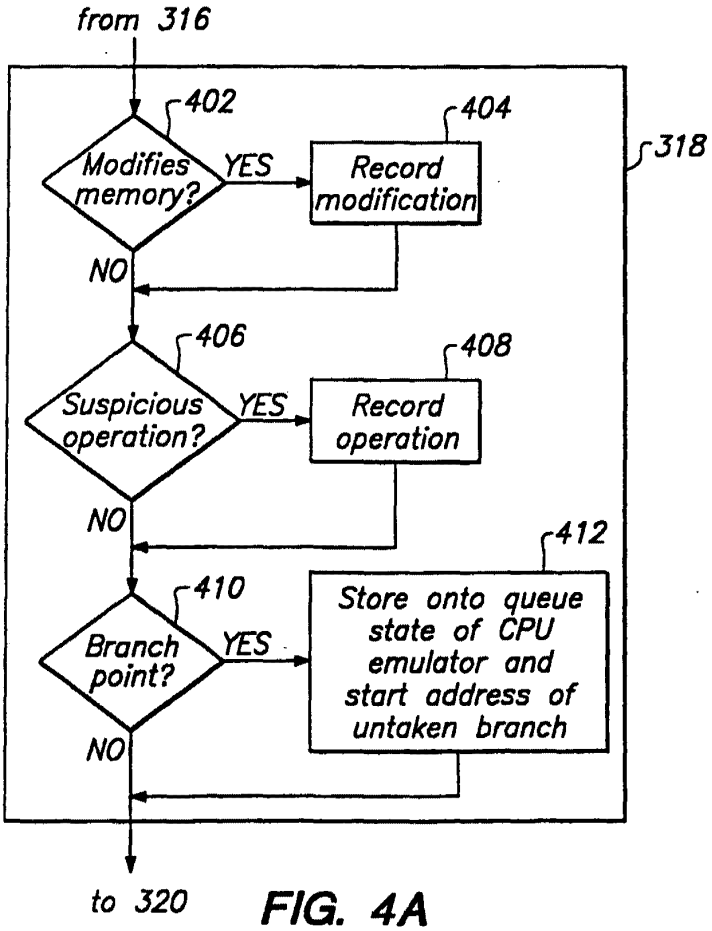

from 316
402
Modifies memory?
YES
404
Record modification
318
NO
406
Suspicious operation?
YES
408
Record operation
NO
412
410
Branch point?
YES
Store onto queue state of CPU emulator and start address of untaken branch
NO
to 320

FIG. 4A

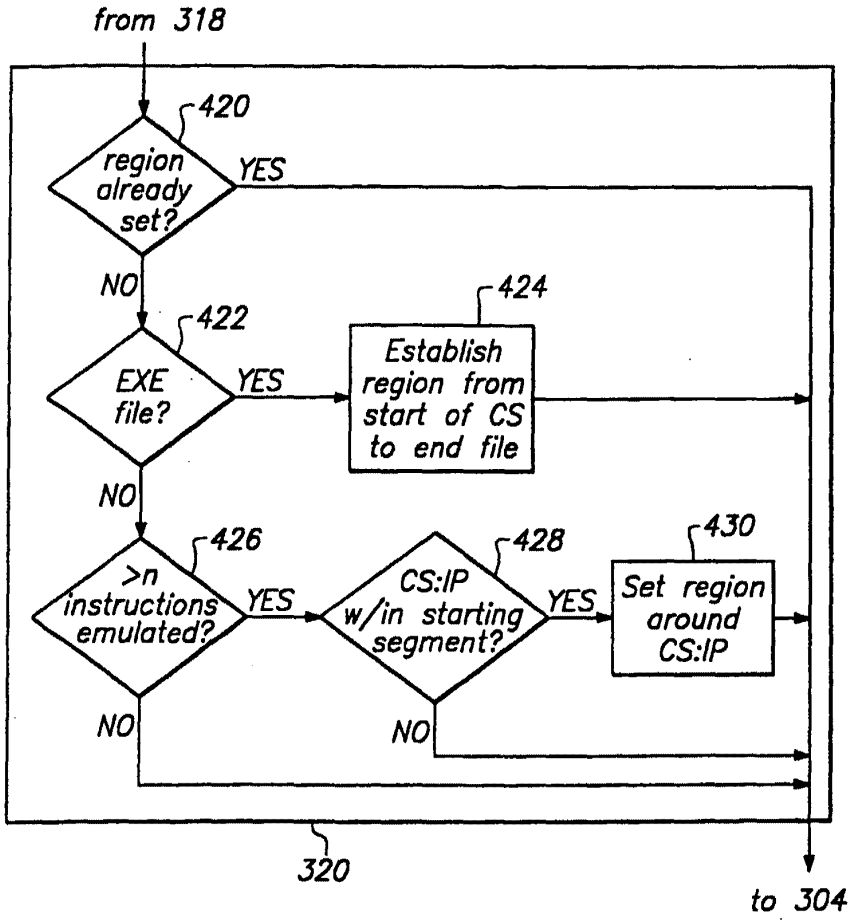

from 318
420
region already set?
YES
NO
422
EXE file?
YES
424
Establish region from start of CS to end file
NO
426
>n instructions emulated?
YES
428
CS:IP w/in starting segment?
YES
430
Set region around CS:IP
NO
NO
320
to 304

FIG. 4B

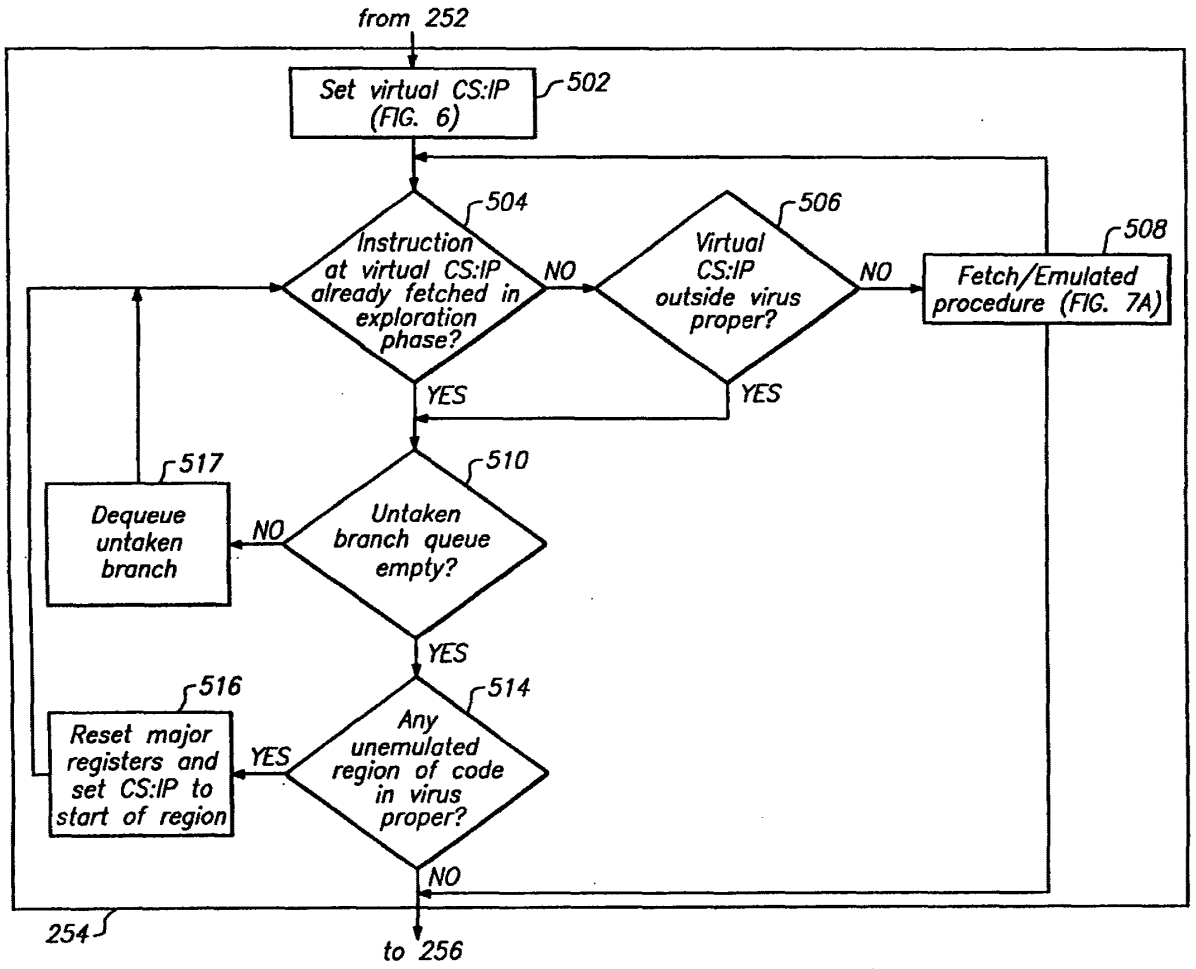


FIG. 5

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1085

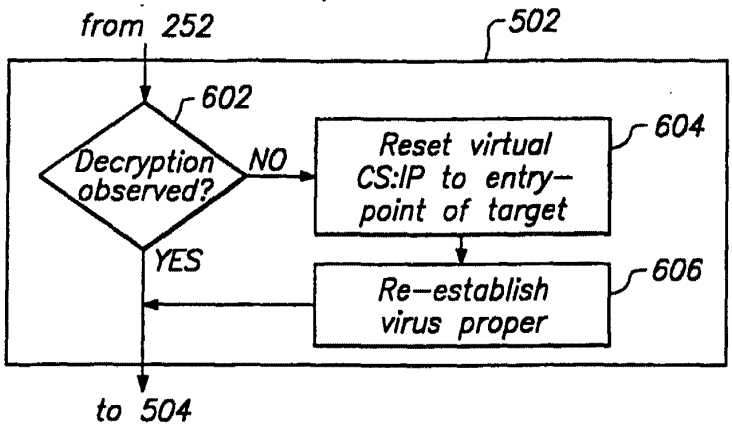

from 252
502
602
Decryption observed?
NO
Reset virtual CS:IP to entry-point of target
604
YES
606
Re-establish virus proper
to 504

FIG. 6

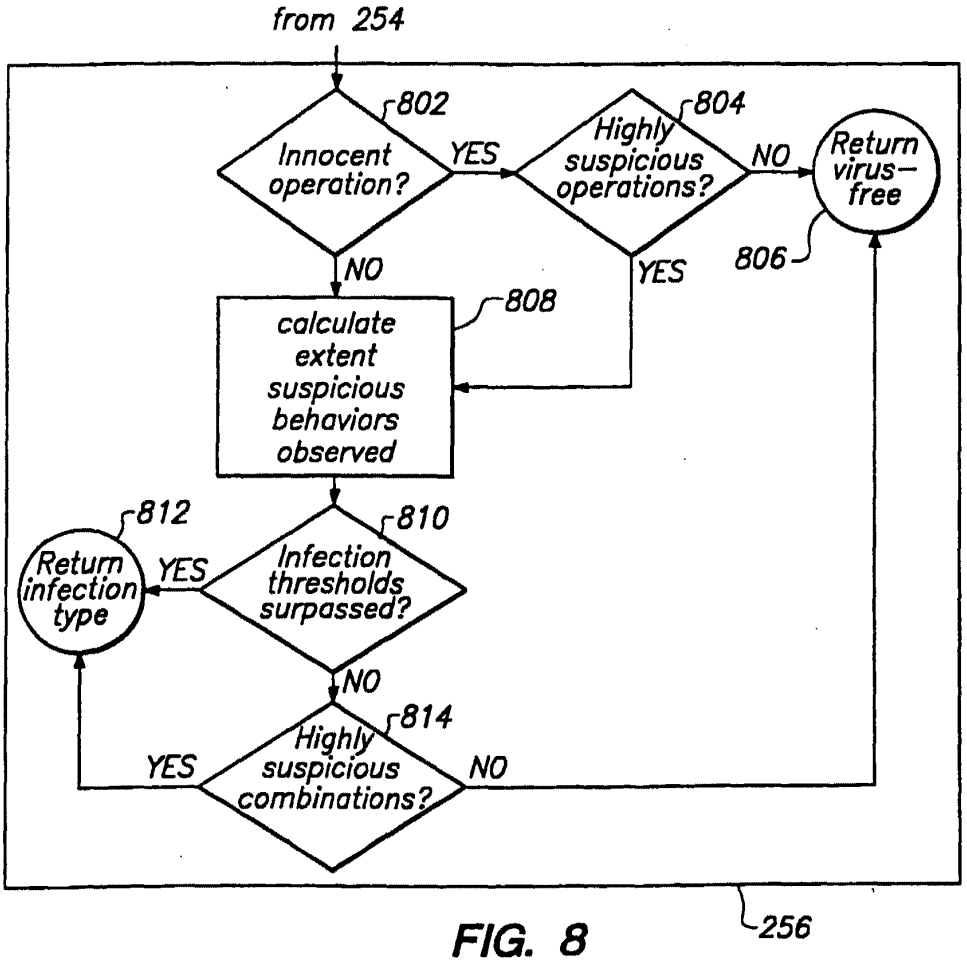

from 254
802
Innocent operation?
YES
804
Highly suspicious operations?
NO
Return virus-free
806
NO
YES
808
calculate extent suspicious behaviors observed
812
Return infection type
YES
810
Infection thresholds surpassed?
NO
814
Highly suspicious combinations?
YES
NO
256

FIG. 8

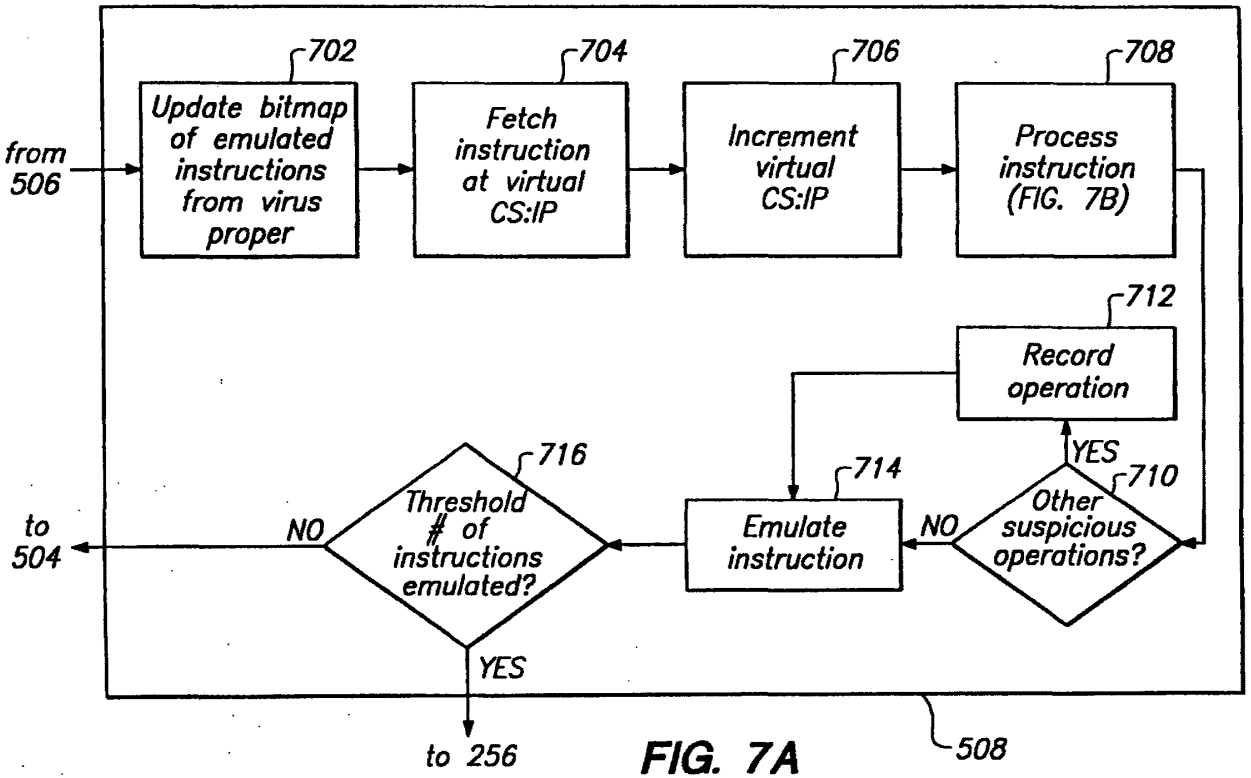

from 506
702
Update bitmap of emulated instructions from virus proper
704
Fetch instruction at virtual CS:IP
706
Increment virtual CS:IP
708
Process instruction (FIG. 7B)
712
Record operation
YES
710
Other suspicious operations?
NO
714
Emulate instruction
716
Threshold # of instructions emulated?
NO
to 504
YES
to 256
508

FIG. 7A

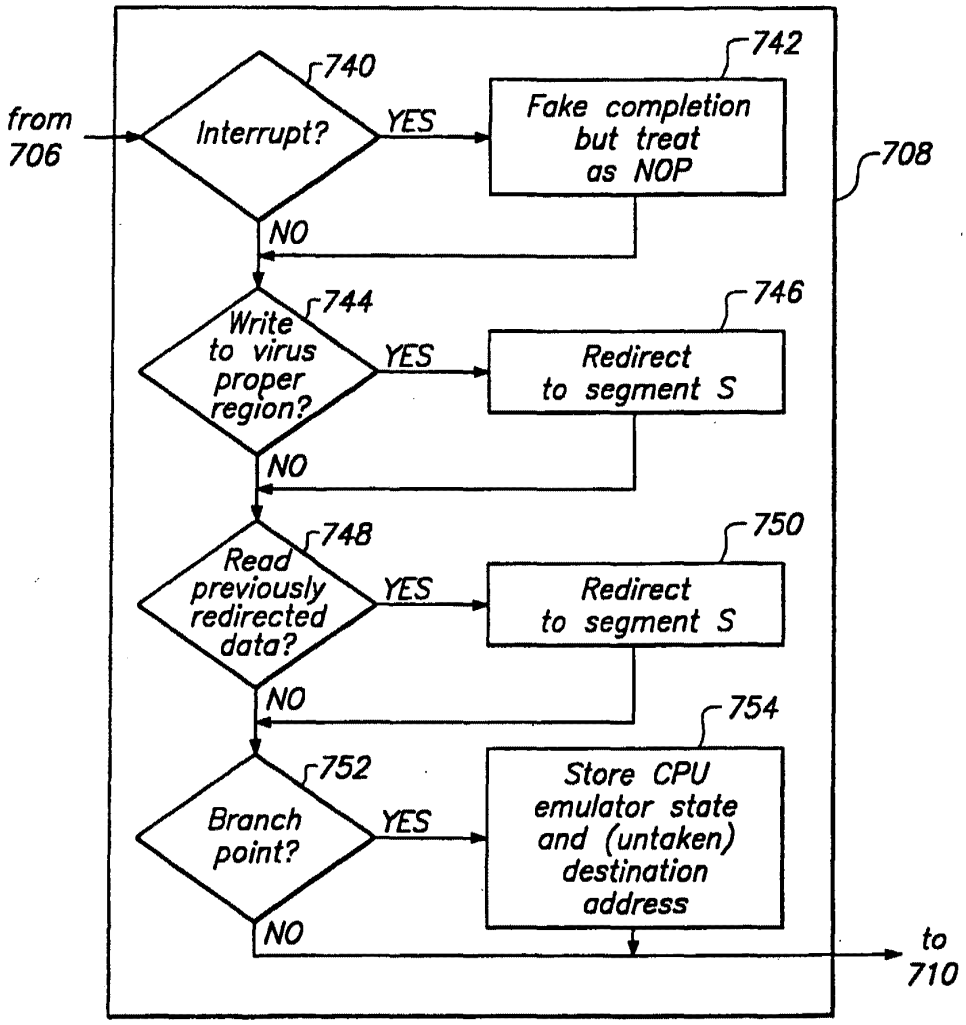

from 706
740
Interrupt?
YES
742
Fake completion but treat as NOP
708
NO
744
Write to virus proper region?
YES
746
Redirect to segment S
NO
748
Read previously redirected data?
YES
750
Redirect to segment S
NO
754
752
Branch point?
YES
Store CPU emulator state and (untaken) destination address
NO
to 710

FIG. 7B

INTERNATIONAL SEARCH REPORT

Interna .l Application No
PCT/US 98/17609

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G06F11/00

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 6 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)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CAREY STOVER NACHENBERG: "A New Technique For Detecting Polymorphic Computer Viruses" 1995 , UNIVERSITY OF CALIFORNIA , LOS ANGELES XP000197628 see page 70, line 1 - page 82, line 6	1-5, 8-11, 13-20,36
Y	---	6,7,12
X	EP 0 636 977 A (DAVID ALAN CHAMBERS) 1 February 1995 see column 8, line 15 - column 12, line 58	1
Y		6,7,12
A	see column 14, line 5 - line 17	21-35, 38-46
	--- -/--	

[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
12 January 1999	21/01/1999

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	Corremans, G

1

Form PCT/ISA/210 (second sheet) (July 1992)

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1089

INTERNATIONAL SEARCH REPORT

Interna l Application No
PCT/US 98/17609

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
A	WO 95 33237 A (QUANTUM LEAP INNOVATIONS INC.) 7 December 1995 see page 12, line 23 - page 13, line 24	

1

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1090

INTERNATIONAL SEARCH REPORT

Information on patent family members

Interna al Application No
PCT/US 98/17609

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 636977 A	01-02-1995	US 5398196 A	14-03-1995
WO 9533237 A	07-12-1995	CA 2191205 A	07-12-1995
		EP 0769170 A	23-04-1997
		JP 10501354 T	03-02-1998
		US 5842002 A	24-11-1998

Form PCT/ISA/210 (patent family annex) (July 1992)

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1091

WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Thursday, September 28, 2006

Hide?	Set Name	Query	Hit Count
	DB=PGPB,USPT,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR		
☐	L19	L18 and checksum	40
☐	L18	(virus or hostile or malicious or trojan adj horse or suspicious) and L17	134
☐	L17	(list or database)near5(allow? or approv? or pre-approv)near5(files or programs or executables)	1581
☐	L16	L15 and (signature)near5(database or list)	46
☐	L15	(malware or virus)near5(scan?)	393
☐	L14	L13 and Kernel	110
☐	L13	L12 and checksum	550
☐	L12	L11 and L4	4471
☐	L11	(allow?)near5(programs or software or signatures)	58206
☐	L10	(pre-approv?)near5(software or programs or signatures)	1
☐	L9	(pre-approv?)near5(files or executables)	3
☐	L8	L6 and L7	12
☐	L7	(authenticat? or validat? or permit? or authoriz? or verif?)near5(new)near5 (program or executable or software or executables or signature or signatures)	415
☐	L6	(new)near5(program or executable or software or executables or signature or signatures)near5(run? or execut?)	974
☐	L5	L3 and L4	4491
☐	L4	(permit? or validat? or authenticat? or authoriz?)near5(program or executable or software or executables or signature or signatures)	18837
☐	L3	(pre-approv? or allow?)near5(program or executable or software or executables or signature or signatures)	58632
☐	L2	L1 and checksum	90
☐	L1	Virus adj signature	406

END OF SEARCH HISTORY

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1092

WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Friday, September 29, 2006

Hide?	Set Name	Query	Hit Count
	DB=PGPB,USPT,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR		
☐	L14	713/167.ccls.	318
☐	L13	713/188.ccls. and (pre-approv? or approv? or allow?)near5(program or executable or software or executables or signature or signatures)	58
☐	L12	713/187.ccls. and (pre-approv? or approv? or allow?)near5(program or executable or software or executables or signature or signatures)	59
☐	L11	713/182.ccls. and (pre-approv? or approv? or allow?)near5(program or executable or software or executables or signature or signatures)	134
☐	L10	713/165.ccls.	550
☐	L9	713/164.ccls.	361
☐	L8	713/161.ccls. and (pre-approv? or approv? or allow?)near5(program or executable or software or executables or signature or signatures)	31
☐	L7	726/26.ccls.	1016
☐	L6	726/24.ccls.	348
☐	L5	726/22.ccls.	676
☐	L4	726/14.ccls.	270
☐	L3	726/14.ccls. and (pre-approv? or approv? or allow?)near5(program or executable or software or executables or signature or signatures)	33
☐	L2	726/2.ccls. and (pre-approv? or approv? or allow?)near5(program or executable or software or executables or signature or signatures)	46
☐	L1	726/1.ccls. and (pre-approv? or approv? or allow?)near5(program or executable or software or executables or signature or signatures)	84

END OF SEARCH HISTORY

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1093

WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Friday, September 29, 2006

Hide?	Set Name	Query	Hit Count
	DB=PGPB,USPT,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR		
☐	L22	(intrus? or intrud? or malware or virus or hostile or malicious) and L17	9
☐	L21	L18 and (pre-approv? or approv? or allow?)near5(program or executable or software or executables or signature or signatures)	0
☐	L20	L17 and (pre-approv? or approv? or allow?)near5(program or executable or software or executables or signature or signatures)	0
☐	L19	L15 and L17	3
☐	L18	Hackenberger.inv.	54
☐	L17	Satterlee.inv.	123
☐	L16	L15 and (pre-approv? or approv? or allow?)near5(program or executable or software or executables or signature or signatures)	34
☐	L15	Internet adj security adj system	200

END OF SEARCH HISTORY

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1094

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.
10/336,299	01/03/2003	Thomas James Satterlee	05456.105038	7850

20786 7590 10/11/2006

KING & SPALDING LLP
1180 PEACHTREE STREET
ATLANTA, GA 30309

EXAMINER	
JACKSON, JENISE E	

ART UNIT	PAPER NUMBER
2131	

DATE MAILED: 10/11/2006

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

PTO-90C (Rev. 10/03)

Office Action Summary	Application No.	Applicant(s)	
	10/336,299	SATTERLEE ET AL.	
	Examiner	Art Unit	
	Jenise E. Jackson	2131	

-- 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 *23 January 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) *1-40* 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) *1-40* 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 *20060928*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1096

DETAILED ACTION

Claim Rejections - 35 USC § 112

1. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

2. Claims 26-35 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter, which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. More specifically, independent claim 26, discloses, "determining whether the file is restricted, taking remedial action to protect the computing device; and if the file is not restricted, permitting the call to the open the file". The specification does not disclose the limitations of claim 26. Thus, Claims 26-35 are rejected under 112 1st.

3. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. Claims 26-35 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Claims 26-35 are not described or disclosed in the specification. The specification cannot be amended to include the limitations, which are not disclosed and described in the specification. The limitations of "restricting", and the "remedial actions", that are described in claims 26-35. Thus, the specification cannot be amended to include these features,

because that would constitute new matter. Thus, claims 26-35 should be cancelled to avoid having new matter issues.

Claim Rejections - 35 USC § 102

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

6. Claims 1-40 are rejected under 35 U.S.C. 102(e) as being anticipated by Mateev et al(2003/0101381).

7. As per claim 1, Mateev et al. discloses computer-implemented method for implementing security for a computing device(pg. 4, 0039, 0040), identifying an allowed program that is permitted to execute on the computing device(see pg. 9, 0082-0083); receiving a signal that a new program is going to be executed on the computing device(see pg. 9, 0082-0083, 0087); suspending the execution of the new program on the computing device; determining whether the new program is the same as the allowed program(pg. 9 0083); if the new program is the same as the allowed program, permitting the new program to execute on the computing device; and if the new program is not the same as the allowed program, monitoring the new program while allowing it to execute on the computing device(see pg. 9, 0086-0087).

8. As per claim 2, Mateev discloses wherein the step of receiving a signal that a new

program is going to be executed is performed at the operating system kernel of the computing device(pg. 5 0054, pg. 9 0083).

9. As per claim 3, Mateev discloses wherein the step of monitoring the new program while allowing it to execute is performed at the operating system kernel of the computing device(pg. 5, 0054, pg. 9 0083-0084).

10. As per claim 4, Mateev discloses wherein the step of determining whether the new program is the same as the allowed program(pg. 9, 0082-0083, 0087) computing a checksum for the new program, and comparing the new program's checksum to a checksum for the allowed program(pg. 12 0106).

11. As per claim 5, Mateev discloses wherein the step of determining whether the new program is the same as the allowed program comprises determining whether the new program has been modified (pg. 9 0082-0083, 0087).

12. As per claim 6, Mateev discloses wherein the step of monitoring the new program while allowing it to execute includes controlling the files the program attempts to access during execution of the new program on the computing device(pg. 9 0082-0083, 0087).

13. As per claim 7, Mateev discloses wherein the step of monitoring the new program while allowing it to execute includes controlling the registry settings the new program attempts to access during execution of the new program on the computing device(pg. 7 0066, 0068).

14. As per claim 8, Mateev discloses wherein the step of monitoring the new program while allowing it to execute includes controlling the network activity of the new program(pg. 0082-0083, 0087).

15. As per claim 9, Mateev discloses wherein the step of monitoring the new program while

allowing it to execute includes determining whether a user has previously approved operation of the new program(pg. 9, 0083, pg. 11 0103).

16. As per claim 10, Mateev discloses the step of terminating the execution of the new program if the program performs suspicious activities(pg. 9 0084-0085).

17. As per claim 11, Mateev discloses the step of taking remedial measures to protect the security of other computing device components if the new program is not the same as the allowed program(pg. 9 0082-0083).

18. As per claim 12, Mateev discloses a computer-readable medium having computer-executable instructions for performing the steps recited in claim 1(pg. 4 0039).

19. As per claim 13, Mateev discloses a computer-implemented method for implementing security for a computing device(pg. 4, 0039, 0040), interrupting the loading of a new program for operation with the computing device; validating the new program; if the new program is validated, permitting the new program to continue loading and to execute in connection with the computing device; if the new program is not validated, monitoring the new program while it loads and executes in connection with the computing device(see pg. 9, 0083, 0086-0087).

20. As per claim 14, Mateev discloses wherein the step of interrupting the loading of a new program comprises: intercepting a signal from the computing device's operating system that the new program is loading, and suspending the loading of the new program(pg. 9 0083).

21. As per claim 15, Mateev discloses wherein the step of monitoring the new program includes intercepting a signal from the computing device's operating system(pg. 0082-0083, 0087).

22. As per claim 16, Mateev discloses wherein the step of validating the new program

includes determining whether the new program corresponds with an approved program(pg. 9, 0083, 0086-0087).

23. As per claim 17, Mateev discloses wherein the step of validating the new program includes comparing a checksum for the new program with a previously determined checksum(pg. 12 0106).

24. As per claim 18, Mateev discloses wherein the step of validating the new program comprises analyzing characteristics of the new program(pg. 9 0084-0085).

25. As per claim 19, Mateev discloses wherein the step of monitoring the new program includes controlling the files the new program attempts to access during execution of the new program(pg. 9 0083).

26. As per claim 20, Mateev discloses a computer-readable medium having computer-executable instructions for performing the steps recited in claim 13(pg. 4 0039).

27. As per claim 21, Mateev discloses a computer-implemented method for performing security for a computing device during a pre-execution phase(pg. 9 0083) identifying an allowed program that is permitted to execute with the computing device; receiving a signal that a new program is being loaded for execution with the computing device; suspending the loading of the new program; comparing the new program to the allowed program; and determining whether the new program is valid(pg. 9 0083-0085).

28. As per claim 22, Mateev discloses wherein the step of receiving a signal includes receiving a signal from the computing device's operating system kernel(pg. 5 0054, pg. 9 0083).

29. As per claim 23, Mateev discloses wherein the step of comparing the new program to the allowed program includes comparing a checksum of the new program to a checksum of the

allowed program(pg. 12 0106).

30. As per claim 24, Mateev discloses wherein the step of determining whether the new program is valid includes determining whether a checksum of the new program corresponds to a checksum of the allowed program(pg. 12 0106).

31. As per claim 25, Mateev discloses a computer-readable medium having computer-executable instructions for performing the steps recited in claim 21(pg. 4 0039).

32. As per claim 26, Mateev discloses a computer-implemented method for performing security for a computing device during an execution phase(see pg. 9, 0082-0083), intercepting a call to open a file from the computing device's operating system; determining whether the file is restricted; if the file is restricted, taking remedial action to protect the computing device; and if the file is not restricted, permitting the call to open the file(pg. 11 0099).

33. As per claim 27, Mateev discloses the step of receiving an instruction to monitor a call from an executing program(pg. 11 0099, 0103).

34. As per claim 28, Mateev discloses the step of defining files that are restricted from access(pg. 11, 0105).

35. As per claim 29, Mateev discloses, wherein the step of defining files that are restricted includes designating a file that cannot be accessed in response to the call(pg. 11, 0103).

36. As per claim 30, Mateev discloses wherein the step of defining files that are restricted includes designating a file that cannot be modified in response to the call(pg. 9, 0091).

37. As per claim 31, Mateev discloses the step of alerting a security management module of the presence of a malicious program(pg. 11, 0103-0104).

38. As per claim 32, Mateev discloses wherein the step of taking remedial action includes

terminating the call(pg. 10, 0094).

39. As per claim 33, Mateev discloses wherein the step of taking remedial action includes alerting another security module of the existence of the call(pg. 11, 0104-0105).

40. As per claim 34, Mateev discloses wherein the step of taking remedial action includes notifying the user of the existence of the call(pg. 10, 0094).

41. As per claim 35, Mateev discloses a computer-readable medium having computer-executable instructions for performing the steps recited in claim 26(pg. 4 0039).

42. As per claim 36, Mateev discloses a system for managing security of a computing device a pre-execution module operable for receiving notice from the computing device's operating system that a program is being loaded onto the computing device(pg. 9 0083); a validation module coupled to the pre-execution monitor operable for determining whether the program is valid; a detection module coupled to the pre-execution monitor operable for intercepting a trigger from the computing device's operating system(pg. 9 0083-0085); and an execution module coupled to the detection module and operable for monitoring the program in response to the trigger intercepted by the detection module(see pg. 9, 0086-0087).

43. As per claim 37, Mateev discloses wherein the pre-execution module is further operable for suspending loading of the program onto the computing device(pg. 9 0083-0085).

44. As per claim 38, Mateev discloses wherein the pre-execution module is further operable for retrieving validation data(pg. 11 0103-0104).

45. As per claim 39, Mateev discloses wherein the execution module is further operable for deciding whether to terminate the trigger intercepted by the detection module(pg. 0105-0106).

46. As per claim 40, Mateev discloses wherein the execution module is further operable for

retrieving authorization data and deciding how to respond to the trigger intercepted by the detection module(pg. 12 0107-0108).

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jenise E. Jackson whose telephone number is (571) 272-3791. The examiner can normally be reached on M-Th (6:00 a.m. - 3:30 p.m.) alternate Friday's.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ayaz Sheikh can be reached on (571) 272-3795. 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.

September 29, 2006

AYAZ SHEIKH
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

OIPE IAP2 JAN 23 2006 PATENT AND TRADEMARK OFFICE

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/336,299	January 3, 2003
APPLICANT		**GROUP**
Thomas Satterlee et al.		2131

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	AA	5,696,486	12/9/97	Poliquin et al.	340		
	AB	6,353,385	3/5/02	Molini et al.	340		
	AC	5,353,393	10/4/94	Bennett et al.	345		
	AD	6,429,952	8/6/02	Olbricht	358		
	AE	5,748,098	5/5/98	Grace	370		
	AF	6,396,845	5/28/02	Sugita	370		
	AG	6,711,127	3/23/04	Gorman et al.	370		
	AH	5,734,697	3/31/98	Jabbarnezhad	379		
	AI	6,104,783	8/15/00	DeFino	379		
	AJ	6,226,372	5/1/01	Beebe et al.	379		
	AK	6,445,822	9/3/02	Crill et al.	382		
	AL	6,563,959	5/13/03	Troyanker	382		
	AM	6,618,501	9/9/03	Osawa et al.	382		

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	TRANSLATION NO.

OTHER REFERENCES *(Including Author, Title, Date, Pertinent Pages, Etc.)*

EXAMINER	DATE CONSIDERED
	9/29/06

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1105

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/336,299	January 3, 2003
APPLICANT		**GROUP**
Thomas Satterlee et al.		2131

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	AN	6,628,824	9/30/03	Belanger	382		
	AO	6,647,139	11/11/03	Kunii et al.	382		
	AP	6,661,904	12/9/03	Sasich et al.	382		
	AQ	6,668,082	12/23/03	Davison et al.	382		
	AR	6,668,084	12/23/03	Minami	382		
	AS	6,792,144	9/14/04	Yan et al.	382		
	AT						
	AU	6,278,886	8/21/01	Hwang	455		
	AV	6,889,168	05/03/05	Hartley et al.	702		
	AW	5,210,704	5/11/93	Husseiny	702		
	AX	5,842,002	11/24/98	Schnurer et al.	703		
	AY	5,854,916	12/29/98	Nachenberg	703		
	AZ	5,999,723	12/7/99	Nachenberg	703		
	BA	6,067,410	5/23/00	Nachenberg	703		
	BB	5,864,803	1/26/99	Nussbaum	704		

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	TRANSLATION NO.

OTHER REFERENCES *(Including Author, Title, Date, Pertinent Pages, Etc.)*

EXAMINER	DATE CONSIDERED
	9/28/06

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1106

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/336,299	January 3, 2003
APPLICANT		**GROUP**
Thomas Satterlee et al.		2131

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	BC						
	BD	6,584,454	6/24/03	Hummel, Jr. et al.	705		
	BE	5,675,711	10/7/97	Kephart et al.	706		
	BF	5,907,834	5/25/99	Kephart et al.	706		
	BG	5,649,185	7/15/97	Antognini et al.	707		
	BH						
	BI	5,878,420	3/2/99	de la Salle	707		
	BJ	5,899,999	5/4/99	De Bonet	707		
	BK	6,144,961	11/7/00	de la Salle	707		
	BL	6,161,109	12/12/00	Matamoros et al.	707		
	BM	6,282,546	8/28/01	Gleichauf et al.	707		
	BN	6,463,426	10/8/02	Lipson et al.	707		
	BO	6,647,400	11/11/03	Moran	707		
	BP	4,223,380	9/16/80	Antonaccio et al.	709		

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	TRANSLATION NO.

OTHER REFERENCES *(Including Author, Title, Date, Pertinent Pages, Etc.)*

EXAMINER _____ DATE CONSIDERED 9/28/06

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1107

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/336,299	January 3, 2003
APPLICANT		**GROUP**
Thomas Satterlee et al.		2131

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	BQ	5,309,562	5/3/94	Li	709		
	BR	5,347,450	9/13/94	Nugent	709		
	BS	5,371,852	12/6/94	Attanasio et al.	709		
	BT	5,515,508	5/7/96	Pettus et al.	709		
	BU	5,539,659	7/23/96	McKee et al.	709		
	BV						
	BW	5,706,210	1/6/98	Kumano et al.	709		
	BX	5,745,692	4/28/98	Lohmann, II et al.	709		
	BY	5,790,799	8/4/98	Mogul	709		
	BZ						
	CA						
	CB						
	CC	5,922,051	7/13/99	Sidey	709		
	CD	5,715,395	02/03/98	Brabson et al.	709		
	CE	6,220,768	04/24/01	Barroux	709		

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	TRANSLATION NO.

OTHER REFERENCES *(Including Author, Title, Date, Pertinent Pages, Etc.)*

EXAMINER	DATE CONSIDERED
	9/28/06

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1108

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/336,299	January 3, 2003
APPLICANT		GROUP
Thomas Satterlee et al.		2131

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	CF						
	CG	5,974,237	10/26/99	Shurmer et al.	709		
	CH	5,974,457	10/26/99	Waclawsky et al.	709		
	CH						
	CJ	6,009,467	12/28/99	Ratcliff et al.	709		
	CK						
	CL	6,035,323	3/7/00	Narayen et al.	709		
	CM						
	CN	6,052,709	4/18/00	Paul	709		
	CO	6,070,190	5/30/00	Reps et al.	709		
	CP	6,073,172	6/6/00	Frailong et al.	709		
	CQ						
	CR	6,119,165	9/12/00	Li et al.	709		
	CS	6,195,687	2/27/01	Greaves et al.	709		

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	TRANSLATION NO.

OTHER REFERENCES *(Including Author, Title, Date, Pertinent Pages, Etc.)*

EXAMINER [signature] DATE CONSIDERED 9/28/06

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.

3974796_1.DOC

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1109

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/336,299	January 3, 2003
APPLICANT		**GROUP**
Thomas Satterlee et al.		2131

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	CT	6,377,994	4/23/02	Ault et al.	709		
	CU	6,397,245	5/28/02	Johnson, II et al.	709		
	CV	6,415,321	7/2/02	Gleichauf et al.	709		
	CW	6,434,615	8/13/02	Dinh et al.	709		
	CX	6,438,600	8/20/02	Greenfield et al.	709		
	CY	6,453,345	9/17/02	Trcka et al.	709		
	CZ	6,453,346	9/17/02	Garg et al.	709		
	DA	6,484,203	11/19/02	Porras et al.	709		
	DB	6,496,858	12/17/02	Frailong et al.	709		
	DC	6,519,647	2/11/03	Howard et al.	709		
	DD	6,708,212	3/16/04	Porras et al.	709		
	DE	6,711,615	3/23/04	Porras et al.	709		
	DF	6,718,383	4/6/04	Hebert	709		
	DG	2002/0032793	3/14/02	Malan et al.	709		

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	NO.

OTHER REFERENCES *(Including Author, Title, Date, Pertinent Pages, Etc.)*

EXAMINER	DATE CONSIDERED
	9/28/06

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1110

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/336,299	January 3, 2003
APPLICANT		**GROUP**
Thomas Satterlee et al.		2131

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	DH	2003/0037136	2/20/03	Labovitz et al.	709		
	DI	4,400,769	8/23/83	Kaneda et al.	710		
	DJ	5,511,184	4/23/96	Lin	710		
	DK	6,467,002	10/15/02	Yang	710		
	DL	5,991,856	11/23/99	Spilo et al.	711		
	DM	6,324,627	11/27/01	Kricheff et al.	711		
	DN	4,975,950	12/4/90	Lentz	713		
	DO	5,032,979	7/16/91	Hecht et al.	713		
	DP	5,121,345	6/9/92	Lentz	713		
	DQ	5,204,966	4/20/93	Wittenberg et al.	713		
	DR	5,274,824	12/28/93	Howarth	713		
	DS						
	DT	5,311,593	5/10/94	Carmi	713		
	DU						
	DV						
	DW						

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	NO.

OTHER REFERENCES *(Including Author, Title, Date, Pertinent Pages, Etc.)*

EXAMINER	DATE CONSIDERED 9/28/10

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1111

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/336,299	January 3, 2003
APPLICANT		**GROUP**
Thomas Satterlee et al.		2131

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	DX	5,557,742	9/17/96	Smaha et al.	713		
	DY						
	DZ						
	EA	5,623,600	4/22/97	Ji et al.	713		
	EB						
	EC	5,696,822	12/9/97	Nachenberg	713		
	ED						
	EE						
	EF						
	EG						
	EH						
	EI						
	EJ	5,812,763	9/22/98	Teng	713		
	EK						
	EL	5,822,517	10/13/98	Dotan	713		
	EM	6,851,057	02/01/05	Nachenberg	713		

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	TRANSLATION NO.

OTHER REFERENCES (*Including Author, Title, Date, Pertinent Pages, Etc.*)

EXAMINER	DATE CONSIDERED
	9/28/06

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1112

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/336,299	January 3, 2003
APPLICANT		**GROUP**
Thomas Satterlee et al.		2131

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	EN	5,826,013	10/20/98	Nachenberg	713		
	EO						
	EP						
	EQ						
	ER						
	ES	5,845,067	12/1/98	Porter et al.	713		
	ET	5,848,233	12/8/98	Radia et al.	713		
	EU						
	EV						
	EW	5,892,903	4/6/99	Klaus	713		
	EX	5,919,257	7/6/99	Trostle	713		
	EY	5,919,258	7/6/99	Kayashima et al.	713		
	EZ	5,925,126	7/20/99	Hsieh	713		
	FA	5,931,946	8/3/99	Terada et al.	713		
	FB	5,940,591	8/17/99	Boyle et al.	713		
	FC	6,886,102	04/26/05	Lyle	713		

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	TRANSLATION NO.

OTHER REFERENCES *(Including Author, Title, Date, Pertinent Pages, Etc.)*

EXAMINER	DATE CONSIDERED 9/28/06

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.

3974796_1.DOC

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1113

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/336,299	January 3, 2003
APPLICANT		**GROUP**
Thomas Satterlee et al.		2131

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	FD	5,961,644	10/5/99	Kurtzberg et al.	713		
	FE	5,978,917	11/2/99	Chi	713		
	FF						
	FG	5,983,350	11/9/99	Minear et al.	713		
	FH						
	FI						
	FJ						
	FK	5,991,881	11/23/99	Conklin et al.	713		
	FL	5,999,711	12/7/99	Misra et al.	713		
	FM	6,003,132	12/14/99	Mann	713		
	FN						
	FO	6,070,244	5/30/00	Orchier et al.	713		
	FP	6,081,894	6/27/00	Mann	713		
	FQ	6,493,752	12/10/02	Lee et al.	713		
	FR	5,905,859	05/18/99	Holloway et al.	713		

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	TRANSLATION NO.

OTHER REFERENCES *(Including Author, Title, Date, Pertinent Pages, Etc.)*

EXAMINER	DATE CONSIDERED
	9/28/06

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1114

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/336,299	January 3, 2003
APPLICANT		**GROUP**
Thomas Satterlee et al.		2131

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	FS						
	FT	6,088,804	7/11/00	Hill et al.	713		
	FU	6,092,194	7/18/00	Touboul	713		
	FV						
	FW						
	FX	6,122,738	9/19/00	Millard	713		
	FY	6,154,844	11/28/00	Touboul et al.	713		
	FZ	6,167,520	12/26/00	Touboul	713		
	GA	6,185,689	2/6/01	Todd, Sr. et al.	713		
	GB	6,205,552	3/20/01	Fudge	713		
	GC						
	GD						
	GE	6,272,641	8/7/01	Ji	713		
	GF						
	GG	6,275,942	8/14/01	Bernhard et al.	713		
	GH	6,279,113	8/21/01	Vaidya	713		

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	TRANSLATION NO.

OTHER REFERENCES *(Including Author, Title, Date, Pertinent Pages, Etc.)*

EXAMINER _____ DATE CONSIDERED 9/28/06

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.

3974796_1.DOC

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1115

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/336,299	January 3, 2003
APPLICANT		**GROUP**
Thomas Satterlee et al.		2131

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	GI	6,298,445	10/2/01	Shostack et al.	713		
	GJ	6,301,668	10/9/01	Gleichauf et al.	713		
	GK	6,314,520	11/6/01	Schell et al.	713		
	GL	6,321,338	11/20/01	Porras et al.	713		
	GM	6,324,647	11/27/01	Bowman-Amuah	713		
	GN	6,338,141	1/8/02	Wells	713		
	GO	6,347,374	2/12/02	Drake et al.	713		
	GP	6,357,008	3/12/02	Nachenberg	713		
	GQ						
	GR	6,408,391	6/18/02	Huff et al.	713		
	GS	6,460,141	10/1/02	Olden	713		
	GT	6,470,449	10/22/02	Blandford	713		
	GU	6,477,648	11/5/02	Schell et al.	713		
	GV	6,477,651	11/5/02	Teal	713		

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	TRANSLATION NO.

OTHER REFERENCES *(Including Author, Title, Date, Pertinent Pages, Etc.)*

EXAMINER [signature] DATE CONSIDERED 9/28/06

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.

3974796_1.DOC

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1116

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/336,299	January 3, 2003
APPLICANT		**GROUP**
Thomas Satterlee et al.		2131

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	GW	6,487,666	11/26/02	Shanklin et al.	713		
	GX	6,499,107	12/24/02	Gleichauf et al.	713		
	GY	6,510,523	1/21/03	Perlman et al.	713		
	GZ	6,519,703	2/11/03	Joyce	713		
	HA	6,530,024	3/4/03	Proctor	713		
	HB	6,546,493	4/8/03	Magdych et al.	713		
	HC	6,574,737	6/3/03	Kingsford et al.	713		
	HD	6,578,147	6/10/03	Shanklin et al.	713		
	HE	6,681,331	1/20/04	Munson et al.	713		
	HF	6,691,232	2/10/04	Wood et al.	713		
	HG	6,704,874	3/9/04	Porras et al.	713		
	HH	6,725,377	4/20/04	Kouznetsov	713		
	HI	6,725,378	4/20/04	Schuba et al.	713		
	HJ	6,775,780	8/10/04	Muttik	713		

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	TRANSLATION NO.

OTHER REFERENCES *(Including Author, Title, Date, Pertinent Pages, Etc.)*

EXAMINER [signature] DATE CONSIDERED 9/28/06

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.

3974796_1.DOC

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1117

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/336,299	January 3, 2003
APPLICANT		**GROUP**
Thomas Satterlee et al.		2131

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	HK	6,792,546	9/14/04	Shanklin et al.	713		
	HL	6,816,973	11/9/04	Gleichauf et al.	713		
	HM	6,839,850	1/4/05	Campbell et al.	713		
	HN	2001/0034847	10/25/01	Gaul, Jr.	713		
	HO	2002/0035698	3/21/02	Malan et al.	713		
	HP	2002/0083331	6/27/02	Krumel	713		
	HQ	2002/0083334	6/27/02	Rogers et al.	713		
	HR	2002/0138753	9/26/02	Munson	713		
	HS	2002/0144156	10/3/02	Copeland, III	713		
	HT	2003/0088791	5/8/03	Porras et al.	713		
	HU	2003/0212903	11/13/03	Porras et al.	713		
	HV	2004/0010718	1/15/04	Porras et al.	713		
	HW	4,672,609	6/9/87	Humphrey et al.	714		
	HX	4,773,028	9/20/88	Tallman	714		
	HY	5,398,196	3/14/95	Chambers	714		

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	NO.

OTHER REFERENCES (*Including Author, Title, Date, Pertinent Pages, Etc.*)

EXAMINER [signature] DATE CONSIDERED 9/25/06

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.

3974796_1.DOC

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1118

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/336,299	January 3, 2003
APPLICANT		**GROUP**
Thomas Satterlee et al.		2131

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	HZ	5,440,723	8/8/95	Arnold et al.	714		
	IA	5,452,442	9/19/95	Kephart	714		
	IB	5,649,095	7/15/97	Cozza	714		
	IC	5,765,030	6/9/98	Nachenberg et al.	714		
	ID	5,964,889	10/12/99	Nachenberg	714		
	IE	6,006,016	12/21/99	Faigon et al.	714		
	IF						
	IG	6,021,510	2/1/00	Nachenberg	714		
	IH	6,029,256	2/22/00	Kouznetsov	714		
	II						
	IJ	6,094,731	7/25/00	Waldin et al.	714		
	IK	6,108,799	8/22/00	Boulay et al.	714		
	IL	6,173,413	1/9/01	Slaughter et al.	714		
	IM	6,912,676	06/28/05	Gusler et al.	714		

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	TRANSLATION NO.

OTHER REFERENCES *(Including Author, Title, Date, Pertinent Pages, Etc.)*

EXAMINER [signature] DATE CONSIDERED 9/28/06

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.

3974796_1.DOC

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1119

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/336,299	January 3, 2003
APPLICANT		**GROUP**
Thomas Satterlee et al.		2131

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	IN	6,199,181	3/6/01	Rechef et al.	714		
	IO						
	IP	6,314,525	11/6/01	Mahalingham et al.	714		
	IQ	6,324,656	11/27/01	Gleichauf et al.	714		
	IR	6,601,190	7/29/03	Meyer et al.	714		
	IS	2002/0032880	3/14/02	Poletto et al.	714		
	IT	5,454,074	9/26/95	Hartel et al.	715		
	IU	5,522,026	5/28/96	Records et al.	715		
	IV	6,271,840	8/7/01	Finseth et al.	715		
	IW	6,517,587	2/11/03	Satyavolu et al.	715		
	IX	6,535,227	3/18/03	Fox et al.	715		
	IY						
	IZ						
	JA	5,774,727	6/30/98	Walsh et al.	717		
	JB	5,872,978	2/16/99	Hoskins	717		

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	TRANSLATION NO.

OTHER REFERENCES *(Including Author, Title, Date, Pertinent Pages, Etc.)*

EXAMINER	DATE CONSIDERED
	9/28/06

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.

3974796_1.DOC

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1120

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/336,299	January 3, 2003
APPLICANT		**GROUP**
Thomas Satterlee et al.		2131

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	JC						
	JD	6,118,940	9/12/00	Alexander, III et al.	717		
	JE	6,405,364	6/11/02	Bowman-Amuah	717		
	JF	6,606,744	8/12/03	Mikurak	717		
	JG	6,397,242	5/28/02	Devine et al.	718		
	JH	2002/0032717	3/14/02	Malan et al.	718		
	JI	6,477,585	11/5/02	Cohen et al.	719		
	JJ	6,721,806	4/13/04	Boyd et al.	719		

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	TRANSLATION NO.

OTHER REFERENCES *(Including Author, Title, Date, Pertinent Pages, Etc.)*

EXAMINER _____ DATE CONSIDERED 9/29/06

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.

3974796_1.DOC

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1121

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/366,299	January 3, 2003
APPLICANT	GROUP	BOX 1 / 4
Thomas Satterlee et al.	2131	

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	TRANSLATION NO.
	JK						
	JL						
	JM						
	JN						
	JO	0 985 995	08/13/03	EP	International Business Machines Corp.		
	JP						
	JQ	WO 00/02115	01/13/00	PCT	PRC Inc.		
	JR	WO 99/00720	01/07/99	PCT	Telefonaktiebolaget LM Ericsson		
	JS	WO 99/50734	10/07/99	PCT	AT&T Corporation		
	JT	WO 98/41919	09/24/98	PCT	Trend Micro, Inc.		
	JU	WO 99/53391	10/21/99	PCT	Sun Microsystems, Inc.		
	JV	0 636 977	05/23/01	EP	Chambers		
	JW	WO 99/15966	04/01/99	PCT	Symantec Corporation		

OTHER REFERENCES *(Including Author, Title, Date, Pertinent Pages, Etc.)*

JX		Essex, David, E-Sleuths Make Net Safe for E-Commerce, Computerworld, June 2000, pp. 1-2
JY		Newman, David, Intrusion Detection Systems, Data Communications, 1998, pp. 1-9
JZ		International Search Report for PCT/US02/17161 of December 31, 2002
KA		Hyland, et al., Concentric Supervision of Security Applications: A New Security Management Paradigm Computer Security Applications Conference, 1998, pp. 59-68

EXAMINER [signature] DATE CONSIDERED 9/28/06

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1122

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/366,299	January 3, 2003

APPLICANT	GROUP
Thomas Satterlee et al.	2131

BOX 1 / 4

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES NO.

OTHER REFERENCES *(Including Author, Title, Date, Pertinent Pages, Etc.)*

	KB		Koilpillai et al., Recon-A Tool for Incident Detection, Tracking and Response, Darpa Information Survivability Conference and Exposition, 2000, pp. 199-206
	KC		Alves-Foss, J., An Overview of SNIF: A Tool for Surveying Network Information Flow, Network and Distributed System Security, 1995, pp. 94-101
	KD		Mansouri-Samani et al., A Configurable Event Service for Distributed Systems Configurable Distributed Systems, 1996, pp. 210-217
	KE		International Search Report for PCT/US01/13769 of March 8, 2002
	KF		Jagannathan et al., System Design Document: Next-Generation Intrusion Detection Expert Systems (NIDES), Internet Citation, March 9, 1993, XP002136082, pp. 1-66
	KG		Koilpillai, Adaptive Network Security Management, DARPA NGI PI Conference, October 1998, pp. 1-27
	KH		Hiverworld Continuous Adaptive Risk Management, Hiverworld, Inc., 1999-2000, pp. 1-14
	KI		International Search Report for PCT/US02/04989of September 19, 2002
	KJ		International Search Report for PCT/US02/02917 of August 8, 2002
	KK		
	KL		
	KM		
	KN		

EXAMINER [signature] DATE CONSIDERED 9/23/06

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1123

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/366,299	January 3, 2003

APPLICANT	GROUP
Thomas Satterlee et al.	2131

BOX 1 / 4

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	TRANSLATION NO.

OTHER REFERENCES (*Including Author, Title, Date, Pertinent Pages, Etc.*)

	KO		
	KP		
	KQ		
	KR		
	KS		
	KT		
	KU		
	KV		Guha et al., Network Security via Reverse Engineering of TCP Code: Vulnerability Analysis and Proposed Solution, IEEE, March 1996, pp. 603-610
	KW		Garg et al., High Level Communication Primitives for Concurrent Systems, IEEE, 1988, pp. 92-99
	KX		Hastings et al., TCP/IP Spoofing Fundamentals, IEEE, May 1996, pp. 218-224
	KY		Snapp, Signature Analysis and Communication Issues in a Distributed Intrusion Detection System, Master Thesis, University of California, Davis, California, 1991, pp. 1-40
	KZ		Guha et al., Network Security via Reverse Engineering of TCP Code: Vulnerability Analysis and Proposed Solutions, IEEE, July 1997, pp. 40-48

EXAMINER [signature] DATE CONSIDERED 9/28/06

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1124

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/366,299	January 3, 2003

APPLICANT	GROUP
Thomas Satterlee et al.	2131

BOX 1 / 4

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	TRANSLATION NO.

OTHER REFERENCES (*Including Author, Title, Date, Pertinent Pages, Etc.*)

	LA		Djahandari et al., An MBone for an Application Gateway Firewall, IEEE, November 1997, pp. 72-81
	LB		Kim et al., Implementing a Secure Login Environment: A Case Study of Using a Secure Network Layer Protocol, Department of Computer Science, University of Alabama, June 1995, pp. 1-9
	LC		Satyanarayanan, Integrating Security in a Large Distributed System, Acm Transaction on Computer Systems, Vol. 7, No. 3, August 1989, pp. 47-280
	LD		
	LE		
	LF		
	LG		
	LH		
	LI		
	LJ		
	LK		

EXAMINER [signature] DATE CONSIDERED 9/28/06

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1125

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/366,299	January 3, 2003
APPLICANT	**GROUP**	**BOX 1 / 4**
Thomas Satterlee et al.	2131	

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	TRANSLATION NO.

OTHER REFERENCES *(Including Author, Title, Date, Pertinent Pages, Etc.)*

	LV		
	LW		
	LX		Winn Schwartau, "e.Security™-Solving 'Dumb Days' With Security Visualization," e-Security, Inc., Naples, FL 34103, 2000.
	LY		Anita D'Amico, Ph.D., "Assessment of Open e-Security Platform™: Vendor-Independent Central Management of Computer Security Resources," Applied Visions, Inc., 1999.
	LZ		"e.Security™-Open Enterprise Security Management: Delivering an integrated, automated, centrally Managed Solution You Can Leverage Today and Tomorrow," e-Security, Inc., Naples, FL 34102, 1999.
	MA		"e.Security ™-Vision," e-Security, Inc., Naples, FL, 1999.
	MB		"e.Security™ -Administrator Workbench™," e-Security, Inc. Naples, FL, 1999.
	MC		"e.Security™-Fact Sheet," e-Security, Inc., Naples, FL, 1999.
	MD		"e.Security™-Open e-Security Platform™," e-Security, Inc. Naples, FL, 1999.
	ME		Babcock, "E-Security Tackles The Enterprise," July 28, 1999; Inter@ctive Week, www.Zdnet.com.
	MF		Kay Blough, "In Search of More-Secure Extranets," November 1, 1999, www.InformationWeek.com
	MG		Paul H. Desmond, "Making Sense of Your Security Tools," Software Magazine and Wiesner Publishing, www.softwaremag.com, 1999.
	MH		Kay Blough, "Extra Steps Can Protect Extranets," November 1, 1999, www. InformationWeek.com

EXAMINER _____ DATE CONSIDERED 9/28/06

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1126

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/366,299	January 3, 2003
APPLICANT	**GROUP**	**BOX 1 / 4**
Thomas Satterlee et al.	2131	

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	TRANSLATION NO.

OTHER REFERENCES *(Including Author, Title, Date, Pertinent Pages, Etc.)*

	MI		Sean Hao, "Software protects e-commerce - e-Security's product alerts networks when hackers attack," Florida Today, Florida.
	MJ		Scott Weiss, "Security Strategies - E-Security, Inc.," product brief, Hurwitz Group, Inc., March 24, 2000.
	MK		Sean Adee, CISA, "Managed Risk, Enhanced Response -The Positive Impact of Real-Time Security Awareness," Information Systems Control Journal, Vol. 2, 2000
	ML		"Reprint Review - The Information Security Portal - Open e-Security Platform Version 1.0", February 2000, West Coast Publishing, SC Magazine, 1999
	MM		"e.Security - "Introducing the First Integrated, Automated, and Centralized Enterprise Security Management System," white paper, e-Security, Inc., Naples, FL 34102, 1999
	MN		Ann Harrison, "Computerworld - Integrated Security Helps Zap Bugs," Feb. 21, 2000, Computerworld, Vol. 34, No. 8, Framingham, MA.
	MO		Shruti Daté, "Justice Department Will Centrally Monitor Its Systems For Intrusions," April 3, 2000, Post-Newsweek Business Information, Inc., www.gcn.com.
	MP		e.Security™ , website pages (pp 1-83), www.esecurityinc.com, e-Security, Inc., Naples, FL 34103, September 14, 2000.
	MQ		Peter Sommer, "Intrusion Detection Systems as Evidence," Computer Security Research Centre, United Kingdom.
	MR		Musman et al., System or Security Managers Adaptive Response Tool, DARPA Information Survivability Conference and Exposition, January 25, 2000, pp. 56-68
	MS		Gibson Research Corporation Web Pages, Shields Up! - Internet Connection Security Analysis, grc.com/default.htm, Laguna Hills, California, 2000

EXAMINER _____ DATE CONSIDERED 9/28/06

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1127

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/366,299	January 3, 2003

APPLICANT	GROUP	
Thomas Satterlee et al.	2131	BOX 1 / 4

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	TRANSLATION NO.

OTHER REFERENCES (*Including Author, Title, Date, Pertinent Pages, Etc.*)

	MT		Rouse et al., Design and Evaluation of an Onboard Computer-Based Information System fro Aircraft, IEEE Transactions of Systems, Man, and Cybernetics, Vol. SMC-12, No. 4, July/August 1982, pp. 451-463
	MU		Hammer, An Intelligent Flight-Management Aid for Procedure Execution, IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-14, No. 6, November/December 1984, pp. 885-888
	MV		Mann et al., Analysis of User Procedural Compliance in Controlling a Simulated Process, IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-16, No. 4, July/August 1986
	MW		Todd, Signed and Delivered: An Introduction to Security and Authentication, Find Out How the Jave Security API Can Help you Secure your Code, Javaworld, Web Publishing, Inc., San Francisco, December 1, 1998, pp. 1-5
	MX		Arvind, Secure This. Inform, Association for Information and Image Management, Silver Spring, September /October 1999, pp. 1-4
	MY		Stevens, TCP/IP Illustrated, Vol. 1, 1994, pp. 247
	MZ		Lee et al., A Generic Virus Detection Agent on the Internet, IEEE, 30th Annual Hawaii International Conference on System Sciences, 1997, Vol. 4
	NA		Cutler, Inside Windows NT, 1993, Microsoft Press
	NB		Duncan, Advanced MS-Dos, 1986, Microsoft Press
	NC		McDaniel, IBM Dictionary of Computing, 1994, International Business Machines Corporation
	ND		Burd, Systems Architecture, 1998, Course Technology, Second Edition
	NE		Programmer's Guide PowerJ, 1997, Sybase

EXAMINER _____ DATE CONSIDERED 9/28/06

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1128

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/366,299	January 3, 2003

APPLICANT	GROUP
Thomas Satterlee et al.	2131

BOX 1 / 4

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES NO.

OTHER REFERENCES (*Including Author, Title, Date, Pertinent Pages, Etc.*)

	NF		Swimmer et al., Dynamic detection and classification of computer viruses using general behavior patterns, 1995, Proceedings of the Fifth International Virus Bulletin Conference, Boston
	NG		Advanced Virus Detection Technology for the Next Millennium, August 1999, Network Associates, A Network Associates Executive White Paper, pp. 1-14
	NH		Enterprise-Grade Anti-Virus Automation in the 21st Century, June 2000, Symantec, Technology Brief, pp. 1-17
	NI		Kephart et al., Blueprint for a Computer Immune System, 1997, Retrieved from Internet, URL: http//www.research.ibm.com/antivirus/scipapers/kephart/VB97, pp. 1-15
	NJ		Richardson, Enterprise Antivirus Software, February 2000, Retrieved from Internet, URL: http://www.networkmagazine.com/article/nmg20000426S0006, pp. 1-6
	NK		Understanding and Managing Polymorphic Viruses, 1996, Symantec, The Symantec Enterprise Papers, Vol. XXX, pp. 1-13
	NL		Gong, JavaTM Security Architecture (JDK1.2), October 2, 1998, Sun Microsystems, Inc., Version 1.0, pp. i-iv, 1-62
	NM		Softworks Limited VBVM Whitepaper, November 3, 1998, Retrieved from the Internet, URL: http://web.archive.org/web/19981203105455/http://softworksltd.com/vbvm.html, pp. 1-4
	NN		Kephart, A Biologically Inspired Immune System for Computers, 1994, Artificial Life IV, pp. 130-139
	NO		International Search Report for PCT/US01/26804 of March 21, 2002
	NP		Kosoresow et al., Intrusion Detection via System Call Traces, IEEE Software, pp. 35-42, September/October 1997
	NQ		Veldman, Heuristic Anti-Virus Technology, Proceedings, 3rd International Virus Bulletin Conference, pp. 67-76, September 1993

EXAMINER _____ DATE CONSIDERED 9/28/06

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1129

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/366,299	January 3, 2003
APPLICANT	**GROUP**	**BOX 1 / 4**
Thomas Satterlee et al.	2131	

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	TRANSLATION NO.

OTHER REFERENCES (*Including Author, Title, Date, Pertinent Pages, Etc.*)

	NR		Symantec, Understanding Heuristics: Symantec's Bloodhound Technology, Symantec White Paper Series, Vol. XXXIV, pp. 1-14, September 1997
	NS		Nachenberg, A New Technique for Detecting Polymorphic Computer Viruses, A thesis submitted in partial satisfaction of the requirements for the degree Master of Science in Computer Science, University of California Los Angeles, pp. 1-127, 1995
	NT		Microsoft P-Code Technology, http://msdn.microsoft.com/archive/default.asp?url=/archive/en-us/dnarvc/html/msdn_c7pcode2.asp, pp. 1-6, April 1992
	NU		DJGPP COFF Spec, http://delorie.com/djgpp/doc/coff/, pp. 1-15, October 1996
	NV		Natvig, Sandbox Technology Inside AV Scanners, Virus Bulletin Conference, September 2001, pp. 475-488
	NW		Norman introduces a new technique for eliminating new computer viruses, found on Norman's website, file://c:/documents%20and%20settings\7489\local%20settings\temporary%20internet%20files\olk, pp. 1-2, published October 25, 2001, printed from website December 27, 2002
	NX		International Search Report for PCT/US01/19142 of January 17, 2003
	NY		Using the CamNet BBS FAQ, http://www.cam.net.uk/manuals/bbsfaq/bbsfaq.htm, January 17, 1997.
	NZ		Express Storehouse Ordering System, "Accessing ESOS through the Network", http://www-bfs.ucsd.edu/mss/esos/man3.htm, September 3, 1996.
	OA		NASIRE, NASIRC Bulletin #94-10, http://cs-www.ncsl.nist.gov/secalert/nasa/nasa9410.txt, March 29, 1994.
	OB		Packages in the net directory, http://linux4u.jinr.ru/usoft/WWW/www_debian.org/FTP/net.html, March 20, 1997.
	OC		Sundaram, An Introduction to Intrusion Detection, Copyright 1996, published at www.acm.org/crossroads/xrds2-4/intrus.html, pp. 1-12

EXAMINER _____ DATE CONSIDERED 9/28/06

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1130

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/366,299	January 3, 2003
APPLICANT	**GROUP**	**BOX 1 / 4**
Thomas Satterlee et al.	2131	

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES NO.

OTHER REFERENCES (*Including Author, Title, Date, Pertinent Pages, Etc.*)

	OD		Samfat, IDAMN: An Intrusion Detection Architecture for Mobile Networks, IEEE Journal on Selected Areas in Communications, Vol. 15, No. 7, September 1997, pp. 1373-1380
	OE		INFO: Visual Basic Supports P-Code and Native Code Compilation (Q229415), http://support.micorsoft.com/support/kb/articles/Q229/4/15.ASP, pp. 1-2, April 28, 1999
	OF		International Search Report for PCT/US99/29117 of May 2, 2000

EXAMINER [signature] DATE CONSIDERED 9/28/06

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1131

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/366,299	January 3, 2003

APPLICANT	GROUP	
Thomas Satterlee et al.	2131	BOX 1 / 4

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	TRANSLATION NO.
	OG	WO 99/13427	3/18/99	PCT	MCI Worldcom, Inc.		
	OH	WO 99/57626	11/11/99	PCT	Acquis Technology, Inc.		
	OI	WO 00/10278	2/24/00	PCT	Cryptek Secure Communications, LLC		
	OJ	WO 00/25214	5/4/00	PCT	Securecoft, Inc.		
	OK	WO 00/25527	5/4/00	PCT	Telefonaktiebolaget LM Ericsson		
	OL	WO 00/34867	6/15/00	PCT	Network Ice Corporation		
	OM	WO 02/101516	12/19/02	PCT	Intruvert Networks, Inc.		

OTHER REFERENCES (*Including Author, Title, Date, Pertinent Pages, Etc.*)

EXAMINER _____ DATE CONSIDERED 9/28/06

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.

3974796_1.DOC

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1132

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/366,299	January 3, 2003
APPLICANT	**GROUP**	**BOX 2 / 4**
Thomas Satterlee et al.	2131	

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES NO.

OTHER REFERENCES (*Including Author, Title, Date, Pertinent Pages, Etc.*)

	ON		Nordin, U of MN OIT Security and Assurance, February 9, 2000
	OO		Internet Security Systems, RealSecure SiteProtector, SAFEsuite Decisions to SiteProtector Migration, August 8, 2003, Pages 1-42
	OP		Internet Security Systems, SAFEsuite Enterprise, SAFEsuite Decisions, 1998
	OQ		Internet Security Systems, SAFEsuite Enterprise, Recognizing the Need for Enterprise Security: An Introduction to SAFEsuite Decisions, August 1998, Pages 1-9
	OR		Internet Security Systems, SAFEsuite Decisions 2.6, Frequently Asked Questions, February 21, 2001, Pages 1-10
	OS		Internet Security Systems, SAFEsuite Decisions Version 1.0, User's Guide, 1998, Pages 1-78
	OT		Porras et al., Emerald: Event Monitoring Enabling Responses to Anomalous Live Disturbances, Computer Science Laboratory, SRI International, Menlo Park, CA, October 1997, Pages 353-365
	OU		Cisco Systems, Empowering the Internet Generation, 1998
	OV		Messmer, Start-Up Puts Hackers on BlackICE, Network World Fusion, http://www.nwfusion.com/cgi-bin/mailto/x/cgi, April 21, 1999, Page 1-2
	OW		NeworkICE Corporation, Can You Explain How Your Product Can Protect a Remote User with a VPN Client?, 1998-1999, Pages 1-2, http://www.webarchive.org/web/20000304071415/advice.networkice.com/advice/support/kb/q000003/default
	OX		Yasin, Start-Up Puts Network Intruders on Ice, http://www.internetweek.com/story/INW19990505S0001, May 5, 1999, Pages 1-2

EXAMINER _____ DATE CONSIDERED 7/28/06

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1133

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/366,299	January 3, 2003
APPLICANT	**GROUP**	**BOX 2 / 4**
Thomas Satterlee et al.	2131	

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	TRANSLATION NO.

OTHER REFERENCES (*Including Author, Title, Date, Pertinent Pages, Etc.*)

	OY		Morency, NetworkWorldFusion, http://nwfusion.com/cgi-bin/mailto/x.cgi, June 28, 1999, Pages 1-2
	OZ		Rogers, Network ICE Touts Security Wares, April 23, 1999, San Mateo, California, http://www.crn.com/showArticle.jhtml?articleID=18829106&flatPage=true, Pages 1-2
	PA		Rogers, Network ICE Signs Resellers, May 26, 1999, San Mateo, California, http://www.crn.com/showArticle.jhtml?articleID=18805302&flatPage=true, Pages 1-2
	PB		Internet Security Systems, I've Been Attacked! Now What?, August 27, 1999, http://www.iss.net/security_center/advice/Support/KB/q000033/default.htm, Pages 1-2
	PC		Internet Security Systems, What is the Format of "Attack-List.CSV"?, August 21, 1999, http://www.iss.net/security_center/advice/Support/KB/q000018/default.htm, Pages 1-2
	PD		Neumann et al., Experience with Emerald to Date, April 11-12, 1999, 1st USENIX Workshop on Intrusion Detection and Network Monitoring, Santa Clara, California, Pages 1-9
	PE		Lindqvist et al., Detecting Computer and Network Misuse Through the Production-Based Expert System Toolset (P-BEST), May 9-12, 1999, Proceedings of the 1999 IEEE Symposium on Security and Privacy, Oakland, California, Pages 1-16
	PF		Kendall, A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems, June 1999, Department of Defense Advanced Research Projects Agency, Pages 1-124
	PG		Neumann, Computer Security and the U.S. Infrastructure, November 6, 1997, Congressional Testimony, Pages 1-11
	PH		Porras et al., Life Traffic Analysis of TCP/IP Gateways, November 10, 1997, Internet Society's Networks and Distributed Systems Security Systems Symposium, March 1998, http://www.sdl.sri.com/projects/emerald/live-traffic.html, Pages 1-16

EXAMINER _____ DATE CONSIDERED 9/28/06

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1134

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/366,299	January 3, 2003
APPLICANT	**GROUP**	**BOX 2 / 4**
Thomas Satterlee et al.	2131	

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES NO.

OTHER REFERENCES (*Including Author, Title, Date, Pertinent Pages, Etc.*)

	PI		Raynaud et al., Integrated Network Management IV, 1995, Proceedings of the 4th International Symposium on Integrated Network Management, Pages 1-2 and 5-16
	PJ		Heberlein et al., A Method to Detect Intrusive Activity in a Networked Environment, October 1-4, 1991, 14th National Computer Security Conference, Washington, D.C., Pages 362-363 and 365-371
	PK		Ko et al., Execution Monitoring of Security-Critical Programs in Distributed Systems: A Specification-Based Approach, 1997, Proceedings of the 1997 IEEE Symposium on Security and Privacy, Pages 175-187
	PL		Crosbie et al., Active Defense of a Computer System Using Autonomous Agents, Technical Report No. 95-008, February 15, 1995, Purdue University, West Lafayette, Indiana, Pages 1-14
	PM		Mansouri-Samani et al., Monitoring Distributed Systems, November 1993, IEEE Network, Pages 20-30
	PN		Jakobson et al., Alarm Correlation, November 1993, IEEE Network, Pages 52-59
	PO		Anderson et al., Next-Generation Intrusion Detection Expert (NIDES), A Summary, May 1995, SRI International, Pages 1-37
	PP		Veritas Software, Press Release, Robust Enhancements in Version 6.0 Maintain Seagate WI as the De Facto Standard for Software Distribution, October 6, 1997, Press Releases, Pages 1-4, http://216.239.39.104/search?q=cache:HS9kmK1m2QoJ:www.veritas.com/us/aboutus/pressroom/199....
	PQ		Yasin, Network-Based IDS are About to Stop Crying Wolf, Security Mandate: Silence False Alarms, April 9, 1999, http://lists.jammed.com/ISN/1999/04/0021.html, Pages 1-3
	PR		Internet Security Systems, Press Release, ISS Reports Record Revenues and Net Income for Second Quarter, July 19, 1999, http://bvlive01.iss.net/issEn/delivery/prdetail.jsp?type=Financial&oid=14515, Pages 1-5
	PS		LaPadula, State of the Art in CyberSecurity Monitoring, A Supplement, September 2001, Mitre Corporation, Pages 1-15

EXAMINER _____ DATE CONSIDERED 9/28/06

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1135

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/366,299	January 3, 2003

APPLICANT	GROUP	
Thomas Satterlee et al.	2131	BOX 2 / 4

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	NO.

OTHER REFERENCES (*Including Author, Title, Date, Pertinent Pages, Etc.*)

	PT		Balasubramaniyan et al., An Architecture for Intrusion Detection Using Autonomous Agents, June 11, 1998, Purdue University, West Lafayette, Indiana, Pages 1-4, http://gunther.smeal.psu.edu/images/b9/f3/bb/9e/ba7f39c3871dcedeb9abd0f70cb84607/1.png
	PU		Crosbie et al., Active Defense of a Computer System Using Autonomous Agents, February 15, 1995, Technical Report No. 95-008, Purdue University, West Lafayette, Indiana, Pages 1-14
	PV		Crosbie et al., Defending a Computer System Using Autonomous Agents, March 11, 1994, Technical Report No. 95-022, Purdue University, West Lafayette, Indiana, Pages 1-11
	PW		Denning, An Intrusion-Detection Model, February 1987, IEEE Transactions on Software Engineering, Vol. SE-13, No. 2, Pages 1-17
	PX		Lunt, A Survey of Intrusion Detection Techniques, 1993, Computers & Security, 12 (1993), Pages 405-418
	PY		Porras et al., Penetration State Transition Analysis A Rule-Based Intrusion Detection Approach, 1992, Pages 220-229
	PZ		Javitz et al., The NIDES Statistical Component: Description and Justification, SRI International, Menlo Park, California, SRI Project 3131, March 7, 1994
	QA		Lindqvist et al., Detecting Computer and Network Misuses Through the Production-Based Expert System Toolset (P-BEST), October 25, 1998, Pages 1-20
	QB		Javitz et al., The SRI IDES Statistical Anomaly Detector, SRI Internationa, Menlo Park, California, May 1991, IEEE Symposium on Security and Privacy, Pages 1-11
	QC		Porras et al., Live Traffic Analysis of TCP/IP Gateways, November 10, 1997, SRI International, Menlo Park, California, Pages 1-16

EXAMINER _____ DATE CONSIDERED 9/28/06

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1136

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/366,299	January 3, 2003

APPLICANT	GROUP	
Thomas Satterlee et al.	2131	BOX 2 / 4

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	TRANSLATION NO.

OTHER REFERENCES (*Including Author, Title, Date, Pertinent Pages, Etc.*)

	QD		Porras et al., Live Traffic Analysis of TCP/IP Gateways, December 12, 1997, SRI International, Menlo Park, California, Proceedings of the 1998 ISOC Symposium on Network and Distributed Systems Security, Pages 1-13
	QE		Information & Computing Sciences: System Design Laboratory: Programs: Intrusion Detection, SRI International, http://www.sdl.sri.com/programs/intrusion/, June 17, 2004, Pages 1-2
	QF		Lindqvist et al., eXpert-BSM: A Host-based Intrusion Detection Solution for Sun Solaris, SRI International, Menlo Park, California, December 10-14, 2001, Proceedings of the 17th Annual Computer Security Applications Conference, Pages 1-12
	QG		Almgren et al., Application-Integrated Data Collection for Security Monitoring, October 10-12, 2001, SRI International, Menlo Park, California, Pages 1-15
	QH		Debar et al., Research Report: A Revised Taxonomy for Intrusion-Detection Systems, October 25, 1999, IBM Research, Switzerland, Pages 1-23
	QI		Porras et al., Emerald: Event Monitoring Enabling Responses to Anomalous Live Disturbances, Computer Science Laboratory, SRI International, Menlo Park, CA, December 18, 1996, Pages 1-3.
	QJ		Frequently-Asked Questions about RealSecure, Pages 1-14, http://web.archive.org/web/19970721183227/iss.net/prod/rs_faq.html, May 30, 1997
	QK		Cisco Systems, Inc., Empowering the Internet Generation, 1998.
	QL		Internet Security Systems, Inc., RealSecure Release 1.2 for UNIX, A User Guide and Reference Manual, 1997.
	QM		Internet Security Systems, Inc., Real-time attack recognition and response: A solution for tightening network security, January 1997, Pages 1-13.
	QN		Internet Security Systems, Inc., SAFEsuite Decisions User Guide, Version 1.0, 1998, Pages 1-78

EXAMINER _____ DATE CONSIDERED 9/28/06

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1137

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/366,299	January 3, 2003
APPLICANT	**GROUP**	**BOX 2 / 4**
Thomas Satterlee et al.	2131	

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	TRANSLATION NO.

OTHER REFERENCES *(Including Author, Title, Date, Pertinent Pages, Etc.)*

	QO		NetworkICE Corporation, ICEcap Administrator's Guide, Version 1.0 BETA, 1999, Pages 1-142.
	QP		Debar, Herve et al., A Neural Network Component for an Intrusion Detection System, 1992, Pages 240-250.
	QQ		SRI International, A Prototype IDES: A Real-Time Intrusion-Detection Expert System, August 1987, Page 1-63.
	QR		SRI International, Requirements and Model for IDES-A Real-Time Intrusion-Detection Expert System, August 1985, Pages 1-70.
	QS		SRI International, An Intrusion-Detection Model, November 1985, Pages 1-29.
	QT		Dowell et al., The ComputerWatch Data Reduction Tool, Proc. of the 13th National Computer Security Conference, Washington, D.C., October 1990, Pages 99-108.
	QU		Fox et al., A Neural Network Approach Towards Intrusion Detection, July 2, 1990, Pages 125-134.
	QV		Garvey et al., Model-Based Intrusion Detection, October 1991, Pages 1-14.
	QW		Ilgun et al., State Transition Analysis: A Rule-Based Intrusion Detection Approach, March 1995, Pages 181-199.

EXAMINER [signature] DATE CONSIDERED 9/28/06

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1138

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/366,299	January 3, 2003
APPLICANT	GROUP	BOX 2 / 4
Thomas Satterlee et al.	2131	

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	TRANSLATION NO.

OTHER REFERENCES (*Including Author, Title, Date, Pertinent Pages, Etc.*)

	QX		Javitz et al., The SRI IDES Statistical Anomaly Detector, May 1991, Pages 1-11.
	QY		SRI International, The NIDES Statistical Component Description and Justification, March 7, 1994, Pages 1-46.
	QZ		Karen, Oliver, PC Magazine, The Digital Doorman, November 16, 1999, Page 68.
	RA		Liepins et al., Anomaly Detection: Purpose and Frameowrk, 1989, Pages 495-504.
	RB		Lindqvist et al., Detecting Computer and Network Misuse Through the Production-Bases Expert System Toolset (P-BEST), October 25, 1998, Pages 1-20.
	RC		Lunt, Teresa, A survey of intrusion detection techniques, 1993, Pages 405-418.
	RD		Lunt, Teresa, Automated Audit Trail Analysis and Intrusion Detection: A Survey, October 1988, Pages 1-8.
	RE		Porras et al., Penetration State Transition Analysis A Rule-Based Intrusion Detection Approach, 1992, Pages 220-229.
	RF		Sebring et al., Expert Systems in Intrusion Detection: A Case Study, October 1988, Pages 74-81.
	RG		Shieh et al., A Pattern-Oriented Intrusion-Detection Model and Its Applications, 1991, Pages 327-342.
	RH		Smaha, Stephen, Haystack: An Intrusion Detection System, 1988.
	RI		Snapp, Steven Ray, Signature Analysis and Communication Issues in a Distributed Intrusion Detection System, 1991, Pages 1-40.
	RJ		Porras et al., EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances, October 1997, Pages 353-365.

EXAMINER _____ DATE CONSIDERED 9/28/06

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1139

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/366,299	January 3, 2003
APPLICANT	GROUP	BOX 2 / 4
Thomas Satterlee et al.	2131	

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES NO.

OTHER REFERENCES (*Including Author, Title, Date, Pertinent Pages, Etc.*)

	RK		Lunt et al., Knowledge-Based Intrusion Detection, March 1989, Pages 102-107.
	RL		Lunt et al., An Expert System to Classify and Sanitize Text, December 1987, Pages 1-5.
	RM		Tener, William, Computer Security in the Age of Information, AI & 4GL: Automated Detection and Investigation Tools, 1989, Pages 23-29.
	RN		Teng et al., Adaptive Real-time Anomaly Detection Using Inductively Generated Sequential Patterns, 1990, Pages 278-284.
	RO		Vaccaro et al., Detection of Anomalous Computer Session Activity, 1989, Pages 280-289.
	RP		Winkler, J.R., A UNIX Prototype for Intrusion and Anomaly Detection in Secure Networks, 1990, Pages 115-124.
	RQ		Boyen et al. Tractable Inference for Complex Stochastic Process, July 24-26, 1998.
	RR		Copeland, Observing Network Traffic – Techniques to Sort Out the Good, the Bad, and the Ugly, 2000, Pages 1-7.
	RS		Goan, Terrance, Communications of the ACM, A Cop on the Beat Collecting and Appraising Intrusion Evidence, July 1999, Pages 47-52.
	RT		Heberlein et al., A network Security Monitor, 1990, Pages 296-304.
	RU		Jackson et al., An Expert System Applications for Network Intrusion Detection, October 1991, Pages 1-8.
	RV		Lankewicz et al., Real-Time Anomaly Detection Using a Nonparametric Pattern Recognition Approach, 1991, Pages 80-89.

EXAMINER [signature] DATE CONSIDERED 9/28/06

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1140

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/366,299	January 3, 2003
APPLICANT	GROUP	BOX 2 / 4
Thomas Satterlee et al.	2131	

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	TRANSLATION NO.

OTHER REFERENCES (*Including Author, Title, Date, Pertinent Pages, Etc.*)

	RW		Lippmann et al., Evaluating Intrusion Detection Systems: The 1998 DARPA Off-line Intrusion Detection Evaluation, 1999.
	RX		Munson et al., Watcher: The Missing Piece of the Security Puzzle, December 2001.
	RY		Pearl, Judea, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, September 1988.
	RZ		Porras et al., Live Traffic Analysis of TCP/IP Gateways, December 12, 1997, Pages 1-13.
	SA		EMERALD TCP Statitical Analyzer 1998 Evaluation Results, http://www.sdl.sri.com/projects/emerald/98-eval-estat/index.html, July 9, 1999, Pages 1-15.
	SB		Staniford-Chen, GrIDS-A Graph Based Intrusion Detection System for Large Networks, October 1996.
	SC		Tener, William, Discovery: An Expert System in the Commercial Data Security Environment, December 1986, Pages 45-53.
	SD		Valdes et al., Adaptive, Model-Based Monitoring for Cyber Attack Detection, 2000, Pages 1-19.
	SE		SRI International, Blue Sensors, Sensor Correlation, and Alert Fusion, October 4, 2000.
	SF		Valdes et al., Statistical Methods for Computer Usage Anomaly Detection Using NIDES, January 27, 1995, Pages 306-311.
	SG		Wimer, Scott, The Core of CylantSecure, http://www.cylant.com/products/core.html, 1999, pages 1-4.
	SH		Zhang et al., A Hierarchical Anomaly Network Intrusion Detection System using Neural Network Classification, February 2001.

EXAMINER [signature] DATE CONSIDERED 9/28/06

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1141

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/366,299	January 3, 2003
APPLICANT	**GROUP**	**BOX 2 / 4**
Thomas Satterlee et al.	2131	

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
FOREIGN PATENT DOCUMENTS							
		DOCUMENT NUMBER	DATE	COUNTRY	NAME		TRANSLATION YES NO.

OTHER REFERENCES (*Including Author, Title, Date, Pertinent Pages, Etc.*)

	SI		Cisco Secure Intrusion Detection System 2.1.1 Release Notes, http://www.cisco.com/univercd/cc/td/doc/product/iaabu/csids/csids3/nr211new.htm, June 10, 2003, pages 1-29.
	SJ		Linux Weekly News, http://lwn.net/1998/0910shadow.html, September 8, 1998, pages 1-38.
	SK		Cracker Tracking: Tighter Security with Intrucsion Detection, http://www.bvte.com/art/9805/sec20/art1.htm, May 1998, pages 1-8.
	SL		Cisco Systems, Inc., Newtork RS: Intrusion Detection and Scanning with Active Audit Session 1305, 1998.
	SM		Business Security Advisor Magazine, Intrusion Detection Systems: What You Need to Know, http://advisor.com/doc/0527, September 1998, pages 1-7.
	SN		Garvey et al., An Inference Technique for Integrating Knowledge from Disparate Sources, Multisensor Integration and Fusion for Intelligenct Machines and Systems, 1995, pages 458-464.
	SO		Power et al., CSI Intrusion Detection System Resource, July 1998, pages 1-7.
	SP		Cisco Systems, Inc., NetRanger User's Guide Version 2.1.1, 1998.
	SQ		Internet Security Systems, Real-Time Attack Recognition and Response: A Solution for Tightening Network Security, http://www.iss.net, 1997, Pages 1-13
	SR		Network ICE Corporation, Network ICE Product Documentation, Pages 1-3, http://www.web.archive.org/web/20011005080013/www.networkice.com/support/documentation.html, July 6, 2004
	SS		Network ICE Corporation, Network ICE Documentation, Page 1, http://www.web.archive.org/web/19991109050852/www.networkice.com/support/docs.htm, July 6, 2004
	ST		Network ICE Corporation, Network ICE Press Releases, Page 1, http://www.web.archive.org/web/19990903214428/www.netice.com/company/pressrelease/press.htm, July 7, 2004

EXAMINER _____ DATE CONSIDERED 9/28/06

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1142

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/366,299	January 3, 2003
APPLICANT	GROUP	BOX 2 / 4
Thomas Satterlee et al.	2131	

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	NO.

OTHER REFERENCES (*Including Author, Title, Date, Pertinent Pages, Etc.*)

	SU		Network ICE Corporation, Network ICE Press Releases, Page 1, http://www.web.archive.org/web/20000304074934/www.netice.com/company/pressrelease/press.htm, July 7, 2004
	SV		Brentano et al., An Architecture for Distributed Intrusion Detection System, Department of Energy Computer Security Group, 14th Annual Conference Proceedings, Pages (17)25-17(45), May 1991
	SW		Staniford-Chen et al., GrIDS-A Graph Based Intrusion Detection System for Large Networks, University of California, Davis, California, 19th National Information Systems Security Conference, 1996, Pages 1-10
	SX		Ricciulli et al., Modeling Correlated Alarms in Network Management Systems, SRI International, Menlo Park, California, , Proceedings of the Conference on Communication Networks and Distributed System Modeling and Simulation, 1997, Pages 1-8
	SY		Porras et al., Emerald: Event Monitoring Enabling Responses to Anomalous Live Disturbances, SRI International, Menlo Park, California, 20th National Information Systems Security Conference, October 1997, Pages 1-24
	SZ		Porras et al., A Mission-Impact Based Approach to INFOSEC Alarm Correlation, SRI International, Menlo Park, California, October 2002, Pages 1-33
	TA		Phrack 55 Download (234 kb, 1999-09-09), http://www.phrack.org/show.php?p=55&a=9, Pages 1-6
	TB		Porras et al., A Mission-Impact-Based Approach to INFOSEC Alarm Correlation, SRI International, Menlo Park, California, October 2002, Pages 1-19
	TC		Bace, An Introduction to Intrusion Detection and Assessment for System and Network Security Management, 1999, Pages 1-38

EXAMINER [signature] DATE CONSIDERED 9/28/06

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1143

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/366,299	January 3, 2003

APPLICANT	GROUP	
Thomas Satterlee et al.	2131	BOX 2 / 4

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	NO.

OTHER REFERENCES (*Including Author, Title, Date, Pertinent Pages, Etc.*)

	TD		Hunteman, Automated Information System - (AIS) Alarm System, University of California, Los Alamos National Laboratory, 20th National Information System Security Conference, October 1997, Pages 1-12
	TE		Janakiraman et al., Indra: A Peer-to-Peer Approach to Network Intrusion Detection and Prevention, Proceedings of the 12th International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003, Pages 1-5
	TF		Innella, Intrusion Detection Systems (IDS), Navy Information Assurance, June 28, 2004, Pages 1-4, http://www.infosec.navy.mil/ps/?t=infosecprodsservices/infosecprodsservices.tag&bc=/infosecprodsservices/b...
	TG		Curry, Intrusion Detection Systems, IBM Emergency Response Service, Coast Laboratory, http://www.cerias.purdue.edu/about/history/coast_resources/idcontent/ids.html, July 19, 2004, Pages 1-28
	TH		Lunt et al., Knowledge-Based Intrusion Detection, SRI International, Menlo Park, California, Conference on AI Systems in Government, Washington, D.C., March 1989, Pages 102-107
	TI		A. Information Assurance BAA 98-34 Cover Sheet, SRI International, November 1, 2000, Pages 2-35
	TJ		NetScreen Products, FAQ, http://www.netscreen.com/products/faq.html, February 28, 2003, Pages 1-6
	TK		Miller, A Network Under Attack: Leverage Your Existing Instrumentation to Recognize and Respond to Hacker Attacks, NetScout Systems, Westford, MA, April 2003, Pages 1-8
	TL		Technology Brief: Intrusion Detection for the Millennium, Internet Security Systems, 2000, Pages 1-6
	TM		Weiss et al., Analysis of Audit and Protocol Data using Methods from Artificial Intelligence, Siemens AG, Munich, Germany, Proc. of the 13th National Computer Security Conference, Washington, D.C., October 1990, Pages 109-114
	TN		Snapp et al., DIDS (Distributed Intrusion Detection System) - Motivation, Architecture, and an Early Protype), University of California, Davis California, , Proc. 14th National Computer Security Conference, Washington, D.C., October 1991, Pages 167-176

EXAMINER	DATE CONSIDERED
	9/28/06

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1144

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/366,299	January 3, 2003
APPLICANT	**GROUP**	**BOX 2 / 4**
Thomas Satterlee et al.	2131	

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	TRANSLATION NO.

OTHER REFERENCES *(Including Author, Title, Date, Pertinent Pages, Etc.)*

	TO		Internet Security Systems, Inc., SAFEsuite Enterprise Edition, Project "Gatekeeper" 1.0, Market Requirements Document, March 30, 1998, pp. 1-12
	TP		Internet Security Systems, SAFEsuite Decisions, 2000, pp. 1-2
	TQ		Internet Security Systems, Dynamic Threat Protection, Presse-Roundtable, Munich, Germany, April 10, 2003, pp. 1-63
	TR		Internet Security Systems, Preliminary ODM 1.0 Functional Specification, August 1997, pp. 1-7
	TS		Internet Security Systems, Inc., Scanner-ISSDK Interface, Design Specification, June 2, 2000, Draft 1.07, pp. 1-7
	TT		RealSecure, Adaptive Network Security Manager Module Programmer's Reference Manual, pp. 1-74
	TU		Advanced Concept Technology Demonstrations (ACTD), 2005, pp. 1-28
	TV		Frank, Sounding the Alarm, September 6, 1999, Federal Computer Week, pp. 1-2
	TW		Crumb, Intrusion Detection Systems to be Integrated at AFRL, Air Force Research Laboratory, News@AFRL, Fall 1999, pp. 1
	TX		Temin, Case Study: The IA: AIDE System at Two, 15th Annual Computer Security Applications Conference, December 6-10, 1999, Phoenix, Arizona, pp. 1-26
	TY		Spink, Automated Intrusion Detection Environment (AIDE), Intrusion Detection Sensor Fusion, September 7, 1999, pp. 1-25
	TZ		Frincke et al., A Framework for Cooperative Intrusion Detection, 21st National Information Systems Security Conference, October 6-9, 1998, Crystal City, Virginia, pp. 1-20

EXAMINER [signature] DATE CONSIDERED 9/28/06

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1145

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/366,299	January 3, 2003
APPLICANT	**GROUP**	**BOX 2 / 4**
Thomas Satterlee et al.	2131	

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES NO.

OTHER REFERENCES (*Including Author, Title, Date, Pertinent Pages, Etc.*)

	UA		Anderson et al., In Athena's Camp: Preparing for Conflict in the Information Age, An Exploration of Cyberspace Security R&D Investment Strategies for DARPA: The Day After-in Cyberspace II, Chaper Eleven, 1996, pp. 253-271
	UB		Valdes et al., SRI International, Probabilistic Alert Correlation, 2001, pp. 1-15
	UC		Bass, Multisensor Data Fusion for Next Generation Distributed Intrusion Detection Systems, April 28, 1999, Iris National Symposium, pp. 1-6
	UD		Perrochon et al., Enlisting Event Patterns for Cyber Battlefield Awareness, No Date, Stanford University, pp. 1-12
	UE		Perrochon, Using Context-Based Correlation in Network Operations and Management, July 2, 1999, Stanford University, pp. 1-20
	UF		Perrochon, Real Time Event Based Analysis of Complex Systems, June 1998, pp. 1-9
	UG		Luckham et al., Complex Event Processing in Distributed Systems, August 18, 1988, Stanford University, pp. 1-26
	UH		Pettigrew, US Southcom United States Southern Command's Information Sharing Projects, Summer 1999, IAnewsletter, Vol. 3, No. 1, pp. 1-24
	UI		Levitt et al., CMAD IV, Computer Misuse & Anomaly Detection, Session Summaries, November 12-14, 1996, Proceedings of the Fourth Workshop on Future Directions in Computer Misuse and Anomaly Detection, Monterey, California, pp. 1-86
	UJ		Cheung et al., The Design of GrIDS: A Graph-Based Intrusion Detection System, January 26, 1999, University of California, pp. 1-51

EXAMINER _____ DATE CONSIDERED 9/28/00

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1146

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/366,299	January 3, 2003
APPLICANT	GROUP	BOX 2, 3 / 4
Thomas Satterlee et al.	2131	

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	NO.

OTHER REFERENCES (*Including Author, Title, Date, Pertinent Pages, Etc.*)

	UK		Cohen et al., Report of the Reporting and Analysis Track, Detection of Malicious Code, Intrusions, and Anomalous Activities Workshop, February 22-23, 1999, pp. 1-47
	UL		Garofalakis et al., Network Mining and Analysis: The Nemesis Project, Bell Labs, Lucent Technologies, No Date, pp. 1-12
	UM		RealSecure ICEcap Manager User Guide Version 3.6, Internet Security Systems, Inc., 1998-2002, pp. 1-228
	UN		Cuppens, Cooperative Intrusion Detection, Date Unknown, pp. 1-10
	UO		Mukherjee et al., Network Intrusion Detection, IEEE Network, May/June 1994, pp. 26-41
	UP		Machlis, Net Monitoring Tools Gain Real-Time Alerts, April 14, 1997, http://www.computerworld.com, pp. 1-12
	UQ		OmniGuard/ITA Intruder Alert, AXENT Technologies, Inc., November 17, 2004, http://www.web.archive.org, pp. 1-10
	UR		NetRanger Overview, Chapter 1, Date Unknown, pp. 1-16
	US		Sutterfield, Large-Scale Network Intrusion Detection, 1997, WheelGroup Corporation, pp. 1-10
	UT		Kumar et al., An Application of Pattern Matching in Intrusion Detection, Technical Report No. CSD-TR-94-013, June 17, 1994, Purdue University, pp. 1-55
	UU		Huang et al., A Large-Scale Distributed Intrusion Detection Framework Based on Attack Strategy Analysis, Date Unknown, The Boeing Company, pp. 1-12
	UV		Perrochon et al., Event Mining with Event Processing Networks, Date Unknown, Stanford University, pp. 1-4

EXAMINER | DATE CONSIDERED 9/28/06

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1147

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/366,299	January 3, 2003

APPLICANT	GROUP	
Thomas Satterlee et al.	2131	BOX 3 / 4

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	NO.

OTHER REFERENCES (*Including Author, Title, Date, Pertinent Pages, Etc.*)

	UW		Gruschke, Integrated Event Management: Event Correlation Using Dependency Graphs, presented at DSOM 1998, University of Munich, pp. 1-12
	UX		Bass, Intrusion Detection System and Multisensor Data Fusion, April 2000, Communications of the ACM, Vol. 43, No. 4, pp. 99-105
	UY		Bass et al., A Glimpse into the Future of ID, Date Unknown, Usenix, pp. 1-10
	UZ		LaPadula, State of the Art in Anomaly Detection and Reaction, July 1999, Mitre Corporation, pp. 1-36
	VA		Rationalizing Security Events with Three Dimensions of Correlation, Date Unknown, NetForensics, Tech Brief, pp. 1-6
	VB		Jou et al., Design and Implementation of a Scalable Intrusion Detection System for the Protection of Network Infrastructure, Date Unknown, MCNC, pp. 1-15
	VC		Caldwell, Event Correlation: Security's Holy Grail?, August 1, 2002, GuardedNet, pp. 1-21
	VD		Metcalf et al., Intrusion Detection System Requirements, September 2000, Mitre Corporation, pp. 1-33
	VE		Jou et al., Architecture Design of a Scalable Intrusion Detection System for the Emerging Network Infrastructure, Technical Report CDRL A005, April 1997, MCNC, pp. 1-42
	VF		Security Manager for UNIX Systems Version 3.2.1 User's Guide, February 1998, Internet Security Systems, pp. 1-162
	VG		RealSecure Release 1.2 for UNIX A User Guide and Reference Manual, 1997, Internet Security Systems, Inc., pp. 1-92

EXAMINER [signature] DATE CONSIDERED 9/28/06

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1148

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/366,299	January 3, 2003
APPLICANT	**GROUP**	**BOX 3 / 4**
Thomas Satterlee et al.	2131	

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	TRANSLATION NO.

OTHER REFERENCES *(Including Author, Title, Date, Pertinent Pages, Etc.)*

	VH		Internet Scanner SAFE SAFEsuite 4.0 User Guide and Reference Manual, 1996, Internet Security Systems, Inc., pp. 1-158
	VI		Internet Scanner 3.3 User Guide and Reference Manual, 1996, Internet Security Systems, Inc., pp. 1-119
	VJ		Landwehr et al., Newsletter of the IEEE Computer Society's TC on Security and Privacy Electronics, Electronic CIPHER, November 25, 1997, Issue 25, pp. 1-34
	VK		20th National Information Systems Security Conference, October 6-10, 1997, Baltimore, Maryland, pp. 1-44
	VL		EMERALD Alert Management Interface User's Guide Version 1.2, December 6, 2000, SRI International, pp. 1-11
	VM		Anderson et al., Detecting Unusual Program Behavior Using the Statistical Component of the Next-Generation Intrusion Detection Expert System (NIDES), May 1995, SRI International, pp. 1-89
	VN		Lunt et al., Detecting Intruders in Computer Systems, 1993 Conference on Auditing and Computer Technology, SRI International, pp. 1-17
	VO		Network ICE Products - ICEcap, Date Unknown, pp. 1-2
	VP		Forlanda, The Secrets to Driving on BlackICE, January 12, 2000, Network ICE, pp. 1-35
	VQ		BlackICE User's Guide Version 1.0 Beta Draft, 1999, Network ICE Corporation, pp. 1-59
	VR		ICEcap Administrator's Guide Version 1.0 Beta Draft, 1999, Network ICE Corporation, pp. 1-25
	VS		Shulak et al., ICEcap Advanced Administration Guide Version 3.0, 2001, Internet Security Systems Corporation, pp. 1-51

EXAMINER _____ DATE CONSIDERED 9/28/06

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1149

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/366,299	January 3, 2003
APPLICANT	**GROUP**	**BOX 3 / 4**
Thomas Satterlee et al.	2131	

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	TRANSLATION NO.

OTHER REFERENCES *(Including Author, Title, Date, Pertinent Pages, Etc.)*

	VT		"Real Secure, OS Sensor User Guide," Version 5.0, ©Internet Security Systems, Inc. 199?-2000; September 2000, pgs. 1-64.
	VU		"Real Secure, User's Guide," Version 3.0, © 1992-1998, Internet Security Systems, Inc., pgs. 1-128.
	VV		"System Security Scanner, User Guide," Version 1.6, © 1996-1998, Internet Security Systems, Inc., pgs. 1-164.
	VW		"Real Secure™, Network Engine User Guide," Version 3.2.1, © 1999 by Internet Security Systems, Inc., pgs. 1-38.
	VX		"Real Secure™, User Guide," Version 3.2.1, © 1999 by Internet Security Systems, Inc., pgs. 1-38
	VY		"Real Secure™, Manager for HP OpenView User Guide," Version 1.3, © 1999 by Internet Security Systems, Inc., pgs. 1-48.
	VZ		"Database Scanner, User Guide," Version 2.0, © 1992-1999, Internet Security Systems, Inc., pgs. 1-112.
	WA		"Database Scanner™, User Guide," Version 4.0, © 2000 by Internet Security Systems, Inc., pgs. 1-122.
	WB		"Database Scanner™, User Guide," Version 3.0.1, , © 1999 by Internet Security Systems, Inc., pgs. 1-164.
	WC		"Real Secure™, Network Sensor User Guide," Version 5.0, © 2000 by Internet Security Systems, Inc., pgs. 1-42.
	WD		"Real Secure, Server Sensor User Guide," Version 5.5, © Internet Security Systems, Inc. 2000, pgs. 1-56.
	WE		"Internet Scanner™, User Guide," Version 6.0, Copyright © 1999 by Internet Security Systems, Inc., pgs. 1-182.
	WF		"Internet Scanner™, User Guide," Version 6.1, © 2000 by Internet Security Systems, Inc., pgs. 1-226.
	WG		"Internet Scanner™, User Guide," Version 5.6, © 1992-1998, Internet Security Systems, Inc., pgs. 1-162.

EXAMINER | DATE CONSIDERED 9/28/06

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.

3974796_1.DOC

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1150

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/366,299	January 3, 2003
APPLICANT	GROUP	BOX 4 / 4
Thomas Satterlee et al.	2131	

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	NO.

OTHER REFERENCES *(Including Author, Title, Date, Pertinent Pages, Etc.)*

	WH		"Internet Scanner™, User Guide," Version 5.3, © 1992-1998, Internet Security Systems, Inc. pgs. 1-173.
	WI		"Real Secure, Console User Guide," Version 5.5, © 199?-2000, Internet Security Systems, Inc., pgs. 1-162.
	WJ		"Internet Scanner™, User Guide," Version 5.8, © 1999 by Internet Security Systems, Inc., pgs. 1-148.
	WK		"SAFEsuite Decisions, User Guide," Version 1.0, © 1992-1998, Internet Security Systems, Inc., pgs. 1-88.
	WL		"Real Secure™, Console User Guide," Version 5.0, © 2000 by Internet Security Systems, Inc., pgs. 1-114.
	WM		"SAFEsuite DECISIONS, User Guide," Version 2.5, © 2000 by Internet Security Systems, Inc., pgs. 1-194.
	WN		"System Scanner, User Guide," Version 1.7, © 1992-1998, Internet Security Systems, Inc., pgs. 1-248.
	WO		"Sytem Scanner, User Guide," Version 1.0, © 1996-1998, Internet Security Systems, Inc., pgs. 1-140.
	WP		"System Scanner™, User Guide," Version 4.0, © 1999 by Internet Security Systems, Inc., pgs. 1-178.
	WQ		Internet Security Systems, Inc., "Introduction to RealSecure Version 5.0, The Industry's Only Integrated Host-Based and Network-Based Intrusion Detection System", August 22, 2000, pp. 1-47
	WR		Internet Security Systems, Inc., "RealSecure Network Sensor and Gigabit Network Sensor Policy Guide Version 7.0", August 2003, pp. 1-86
	WS		Internet Security Systems, Inc., "RealSecure Console User Guide Version 3.1", June 1999, pp. 1-98
	WT		Internet Security Systems, Inc., "RealSecure Version 2.0", April 1998, pp. 1- 154
	WU		Internet Security Systems, Inc., "Enhanced Dynamic Threat Protection via Automated Correlation and Analysis", an ISS White Paper, 2002, pp. 1-14

EXAMINER _____ DATE CONSIDERED 9/28/06

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.

3974796_1.DOC

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1151

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/366,299	January 3, 2003

APPLICANT	GROUP
Thomas Satterlee et al.	2131

BOX 4 / 4

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	WV	6,871,284	03/22/05	Cooper et al.	713		
	WW	5,872,915	02/16/99	Dykes et al.	713		

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	TRANSLATION NO.

OTHER REFERENCES (*Including Author, Title, Date, Pertinent Pages, Etc.*)

	WX	Internet Security Systems, Inc., "RealSecure Site Protector Comparison Guide for Internet Scanner 7.0 Reporting Version 1.2", March 2003, an ISS Tech Note, pp. 1-15
	WY	Internet Security System, Inc., "RealSecure Site Protector Comparison Guide for ICEcap Manager Version 1.5", January 2002, an ISS Technical White Paper, pp. 1-27
	WZ	Internet Security Systems, Inc., "RealSecure SiteProtector Security Fusion Module 2.0 Frequently Asked Questions", March 2003, pp. 1-8
	XA	Internet Security Systems, Inc., "RealSecure SiteProtector Console User Reference Guide Version 2.0 Service Pack 1", March 14, 2003, Chapter 3, pp. 1-27
	XB	Internet Security Systems, Inc., "Proventia Gateway and Network Protection", July 2003, pp. 1-12
	XC	Farley, Internet Security System, Inc., "ID Fusion Technology, A First-Generation Approach", April 29, 1999, National Security Framework Forum, pp. 1-8
	XD	Farley, "RealSecure Fusion Engine Concepts", an ISS White Paper, June 30, 1999, pp. 1-23
	XE	McGraw et al., "Secure Computing with Java: Now and the Future", 1997, http://java.sun.com/security/javaone97-whitepaper.html, pgs. 1-20

EXAMINER [signature] DATE CONSIDERED 9/28/06

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.

3974796_1.DOC

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1152

OIPE
MAY 1 4 2003
PATENT & TRADEMARK OFFICE

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/336,299	January 3, 2003
APPLICANT		**GROUP**
Satterlee et al.		

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	AA	5,857,191	01/05/99	Blackwell, Jr. et al.			
	AB	6,035,423	03/07/00	Hodges et al.			
	AC	5,345,595	09/06/94	Johnson et al.			
	AD	6,266,774	07/24/01	Sampath et al.			
	AE	6,026,442	02/15/00	Lewis et al.			
	AF	6,016,553	01/18/00	Schneider et al.			
	AG	6,014,645	01/11/00	Cunningham			
	AH	5,881,236	03/09/99	Dickey			
	AI	5,764,890	06/09/98	Glasser et al.			
	AJ	5,630,061	05/13/97	Richter et al.			
	AK	5,414,833	05/09/95	Hershey et al.			
	AL	4,819,234	04/04/89	Huber			
	AM	6,275,938	08/14/01	Bond et al.			
	AN	5,950,012	09/07/99	Shiell et al.			
	AO	5,278,901	01/11/94	Shieh et al.			
	AP	6,405,318	06/11/02	Rowland			
	AQ	5,475,839	12/12/95	Watson et al.			
	AR	5,590,331	12/31/96	Lewis et al.			
	AS	5,761,504	06/02/98	Corrigan et al.			

EXAMINER _____ DATE CONSIDERED 9/28/06

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.

FORM PTO-1449, Adapted

MAY 1 4 2003

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/336,299	January 3, 2003
APPLICANT		GROUP
Satterlee et al.		

OTHER REFERENCES *(Including Author, Title, Date, Pertinent Pages, Etc.)*

BA		Sammons, Nathaniel, "Multi-platform Interrogation and Reporting with Rscan," The Ninth Systems Administration Conference, LISA 1995, Monterrey, California, September 17-22, 1995, pp. 75-87.
BB		Dean et al., "Java Security: From HotJava to Netscape and Beyond," Proceedings of the 1996 IEEE Symposium on Security and Privacy, May 6-8, 1996, Oakland, California, pp. 190-200.
BC		Fisch et al., "The Design of an Audit Trail Analysis Tool," Proceedings of the 10th Annual Computer Security Applications Conference, December 5-9, 1994, Orlando, Florida, pp. 126-132.
BD		Safford et al., "The TAMU Security Package: An Ongoing Response to Internet Intruders in an Academic Environment," USENIX Symposium Proceedings, UNIX Security IV, October 4-6, 1993, Santa Clara, California, pp. 91-118.
BE		Sugawara, Toshiharu, "A Cooperative LAN Diagnostic and Observation Expert System," Ninth Annual Phoenix Conference on Computers and Communications, 1990 Conference Proceedings, March 21-23, 1990, Scottsdale, Arizona, pp. 667-674.
BF		Casella, Karen A., "Security Administration in an Open Networking Environment," The Ninth Systems Administration Conference, LISA 1995, Monterrey, California, September 17-22, 1995, pp. 67-73.
BG		Burchell, Jonathan, "Vi-SPY: Universal NIM?" Virus Bulletin, January 1995, pp. 20-22.
BH		Benzel et al., "Identification of Subjects and Objects in a Trusted Extensible Client Server Architecture," 18th National Information Systems Security Conference, October 10-13, 1995, Baltimore, Maryland, pp. 83-99.
BI		Epstein et al., "Component Architectures for Trusted Netware," 18th National Information Systems Security Conference, October 10-13, 1995, Baltimore, Maryland, pp. 455-463.
BJ		Varadharajan, Vijay, "Design and Management of a Secure Networked Administration System: A Practical Approach," 19th National Information Systems Security Conference, October 22-25, 1996, Baltimore, Maryland, pp. 570-580.
BK		Snapp et al., "DIDS (Distributed Intrusion Detection System) - Motivation, Architecture, and An Early Prototype," 14th National Computer Security Conference, October 1-4, 1991, Washington, DC, pp. 167-176.
BL		Broner et al., "IntelligentI/O Rule-Based Input/Output Processing for Operating Systems," Operating Systems Review, Volume 25, Number 3, July 1991, pp. 10-26.
BM		Drews et al., "Special Delivery - Automatic Software Distribution Can Make You A Hero," Network Computing, August 1, 1994, pp. 80, 82-86, 89, 91-95.
BN		Morrissey, Peter, "Walls," Network Computing, February 15, 1996, pp. 55-59, 65-67.
BO		Harlander, Dr. Magnus, "Central System Administration in a Heterogenous Unix Environment: GeNUAdmin," Proceedings of the Eighth Systems Administration Conference (LISA VIII), September 19-23, 1994, San Diego, California, pp. 1-8.
BP		Shaddock et al., "How to Upgrade 1500 Workstations on Saturday, and Still Have Time to Mow the Yard on Sunday," The Ninth Systems Administration Conference LISA '95, September 17-22, 1995, Monterrey, California, pp. 59-65.
BQ		Anderson, Paul, "Towards a High-Level Machine Configuration System," Proceedings of the Eighth Systems Administration Conference (LISA VIII), September 19-23, 1994, San Diego, California, pp. 19-26.
BR		Cooper, Michael A., "Overhauling Rdist for the '90s," Proceedings of the Sixth Systems Administration Conference (LISA VI), October 19-23, 1992, Long Beach, California, pp. 175-188.
BS		Vangala et al., "Software Distribution and Management in a Networked Environment," Proceedings of the Sixth Systems Administration Conference, October 19-23, 1992, Long Beach, California, pp. 163-170.
BT		Kim et al., "The Design and Implementation of Tripwire: A File System Integrity Checker," 2nd ACM Conference on Computer and Communications Security, November 2-4, 1994, Fairfax, Virginia, pp. 18-29.

EXAMINER _____ DATE CONSIDERED 9/20/06

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.

FORM PTO-1449 (Adapted)

MAY 16 2003

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038	10/336,299	January 3, 2003
APPLICANT		GROUP
Satterlee et al.		

OTHER REFERENCES (*Including Author, Title, Date, Pertinent Pages, Etc.*)

CA		Mounji et al., "Distributed Audit Trail Analysis," Proceedings of the Symposium on Network and Distributed System Security, San Diego, California, February 16-17, 1995, pp. 102-112.
CB		Wobber et al., "Authentication in the Taos Operating System," ACM Transactions on Computer Systems, Vol. 12, No. 1, February 1994, pp. 3-32.
CC		Mayer et al., "The Design of the Trusted Workstation: A True 'Infosec' Product," 13th National Computer Security Conference, Washington, DC, October 1-4, 1990, pp. 827-839.
CD		Dawson, J. Bruce, "Intrusion Protection for Networks," Byte, April 1995, pp. 171-172.
CE		Buhkan, Michael, "CheckPoint Charlie," PC Week Netweek, November 27, 1995, pp. N1, N6-N7.
CF		Process Software Technical Support page , found on http://www.process.com/techsupport/whitesec.html, printed off of the Process Software website on February 26, 2003, pp. 1-5.
CG		Ganesan, R., "BAfirewall: a modern firewall design," Proceedings Internet Society Symposium on Network and Distributed System Security 1994, Internet Soc. 1994, pp. 99-108, Reston, VA.
CH		Lee, Theodore M.P., "Trusted Systems," Chapter II-1-6 of Handbook of Information Security Management, Ed. Zella G. Ruthberg and Harold F. Tipton, Auerbach, Boston and New York, 1993, pp. 345-362.
CI		Lunt, Teresa F., "Automated Intrusion Detection," Chapter II-4-4 of Handbook of Information Security Management, Ed. Zella G. Ruthberg and Harold F. Tipton, Auerbach, Boston and New York, 1993, pp. 551-563.

FOREIGN PATENT DOCUMENTS

	DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	TRANSLATION NO.
CJ	WO 01/84285	11/08/01	PCT	Internet Security Systems, Inc.		
CK	WO 02/06928	01/24/02	PCT	vCIS, Inc.		
CL	WO 00/54458	09/14/00	PCT	Psionic Software, Inc.		
CM	WO 02/056152	07/18/02	PCT	Psionic Software, Inc.		

EXAMINER _____ DATE CONSIDERED 9/20/06

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.

FORM PTO-1449, Adapted

JUN 25 2003 JC26 PATENT & TRADEMARK OFFICE

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO. 05456.[illegible]	SERIAL NO. 10/336,299	FILING DATE January 3, 2003
APPLICANT Satterlee et al.		GROUP 2131

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	AA	5,359,659	10/25/1994	Doren Rosenthal			6/19/1992
	AB	5,987,610	11/16/1999	Franczek et al.			2/12/1998
	AC	6,230,288 B1	5/8/2001	Kuo et al.			10/29/1998
	AD	6,266,773 B1	7/24/2001	Kisor et al.			12/31/1998
	AE						
	AF						
	AG						
	AH						
	AI						
	AJ						
	AK						

RECEIVED
JUN 26 2003
Technology Center 2100

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	TRANSLATION NO.
	AL	WO 93/25024	5/26/1993	PCT	Reginald K. Branham		
	AM						
	AN						
	AO						

OTHER REFERENCES *(Including Author, Title, Date, Pertinent Pages, Etc.)*

	AP	International Search Report dated May 15, 2003
	AQ	

EXAMINER [signature] DATE CONSIDERED 9/28/06

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.

FORM PTO-1449, Adapted

LIST OF INFORMATION DISCLOSED BY APPLICANT

(Use several sheets if necessary)

JC996 U.S. PTO 10/336299 01/03/03

ATTY. DOCKET NO.	SERIAL NO.	FILING DATE
05456.105038 US	Not Yet Assigned	January 3, 2003

APPLICANT	GROUP
Satterlee et al.	

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	AA						
	AB						
	AC						
	AD						
	AE						
	AF						
	AG						
	AH						
	AI						
	AJ						
	AK						

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	NAME	TRANSLATION YES	TRANSLATION NO.
	AL						
	AM						
	AN						
	AO						

OTHER REFERENCES (*Including Author, Title, Date, Pertinent Pages, Etc.*)

	AP		NXi Communications, Inc.; *White Paper "NTS Security Issues;"* October 15, 2001; pp. 1 - 12
	AR		

EXAMINER _____ DATE CONSIDERED 9/20/06

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.

Notice of References Cited	Application/Control No. 10/336,299	Applicant(s)/Patent Under Reexamination SATTERLEE ET AL.	
	Examiner Jenise E. Jackson	Art Unit 2131	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-2003/0101381	05-2003	Mateev et al.	714/38
*	B	US-2002/0042886	04-2002	Lahti et al.	713/201
*	C	US-2005/0021994	01-2005	Barton et al.	713/200
*	D	US-5,948,104	09-1999	Gluck et al.	726/24
	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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1158

Index of Claims

Application/Control No.	Applicant(s)/Patent under Reexamination
10/336,299	SATTERLEE ET AL.
Examiner	**Art Unit**
Jenise E. Jackson	2131

√	Rejected	–	(Through numeral) Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

Claim		Date
Final	Original	9/29/06
	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	
	32	
	33	
	34	
	35	
	36	
	37	
	38	
	39	
	40	√
	41	
	42	
	43	
	44	
	45	
	46	
	47	
	48	
	49	
	50	

Claim		Date
Final	Original	
	51	
	52	
	53	
	54	
	55	
	56	
	57	
	58	
	59	
	60	
	61	
	62	
	63	
	64	
	65	
	66	
	67	
	68	
	69	
	70	
	71	
	72	
	73	
	74	
	75	
	76	
	77	
	78	
	79	
	80	
	81	
	82	
	83	
	84	
	85	
	86	
	87	
	88	
	89	
	90	
	91	
	92	
	93	
	94	
	95	
	96	
	97	
	98	
	99	
	100	

Claim		Date
Final	Original	
	101	
	102	
	103	
	104	
	105	
	106	
	107	
	108	
	109	
	110	
	111	
	112	
	113	
	114	
	115	
	116	
	117	
	118	
	119	
	120	
	121	
	122	
	123	
	124	
	125	
	126	
	127	
	128	
	129	
	130	
	131	
	132	
	133	
	134	
	135	
	136	
	137	
	138	
	139	
	140	
	141	
	142	
	143	
	144	
	145	
	146	
	147	
	148	
	149	
	150	

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1159

Search Notes

Application/Control No.	Applicant(s)/Patent under Reexamination	
10/336,299	SATTERLEE ET AL.	
Examiner	**Art Unit**	
Jenise E. Jackson	2131	

SEARCHED			
Class	Subclass	Date	Examiner
713	164	9/29/2006	JJ
713	165	9/29/2006	JJ
713	167	9/29/2006	JJ
726	14	9/29/2006	JJ
" "	22	9/29/2006	JJ
" "	24	" "	" "
" "	26	" "	" "

INTERFERENCE SEARCHED			
Class	Subclass	Date	Examiner

SEARCH NOTES (INCLUDING SEARCH STRATEGY)		
	DATE	EXMR
West text search, see printout	9/29/2006	JJ
West class search with text, see printout	9/29/2006	JJ
Inventor and Assignee search, no double patenting, see printout	9/29/2006	JJ

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1160

KING & SPALDING, LLP

RECEIVED
CENTRAL FAX CENTER
JAN 11 2007

1180 Peachtree Street, NE
Atlanta, Georgia 30309-3521
Telephone: 404/572-4600
Facsimile: 404/572-5100
www.kslaw.com

FAX TRANSMITTAL SHEET

January 11, 2007

TO:	Examiner Jenise Jackson GAU 2131 U.S. Serial No. 10/336,299		**Company:**	U.S. Patent and Trademark Office
	Fax #:	**Mail Stop Amendment** 571-273-8300	**City/State:**	Alexandria, VA 22313
FROM:	W. Scott Petty WSP	0252	**Our Ref. #:**	05456.105038

NUMBER OF PAGES (including transmittal sheet): 10

CONFIDENTIALITY NOTICE

THE INFORMATION CONTAINED IN THIS FACSIMILE MESSAGE IS PRIVILEGED AND CONFIDENTIAL INFORMATION INTENDED FOR THE USE OF THE ADDRESSEE LISTED ABOVE. IF YOU ARE NEITHER THE INTENDED RECIPIENT NOR THE EMPLOYEE OR AGENT RESPONSIBLE FOR DELIVERING THIS MESSAGE TO THE INTENDED RECIPIENT, YOU ARE HEREBY NOTIFIED THAT ANY DISCLOSURE, COPYING, DISTRIBUTION OR THE TAKING OF ANY ACTION IN RELIANCE ON THE CONTENTS OF THIS TELECOPIED INFORMATION IS STRICTLY PROHIBITED. IF YOU HAVE RECEIVED THIS TELECOPY IN ERROR, PLEASE IMMEDIATELY NOTIFY US BY TELEPHONE TO ARRANGE FOR RETURN OF THE ORIGINAL DOCUMENTS TO US.

If transmission problems occur or you are not the intended recipient, please call 404.572.2459 immediately. Thank you.

Notes/Comments:

Documents Submitted Via Facsimile:

Applicant:	Thomas James Satterlee et al.
Serial No.:	10/336,299
Filed:	January 3, 2003
For:	System and Method for the Managed Security Control of Processes of a Computer
Papers Faxed:	Amendment Transmittal in duplicate (2-pgs.); Amendment and Response (7-pgs.)

4395213 v1

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1161

PATENTS

RECEIVED
CENTRAL FAX CENTER
JAN 11 2007

IN THE U.S. PATENT AND TRADEMARK OFFICE

In re application of: **Thomas James Satterlee et al.**

Docket No. **05456.105038**

Serial No.: **10/336,299**

Art Unit **2131**

Filed: **January 3, 2003**

For: **System and Method for the Managed Security Control of Processes of a Computer**

Examiner: **Jackson, Jenise E.**

Confirmation No.: **7850**

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

January 11, 2007

Sir:

Transmitted herewith are papers in the above-identified application.

- ☒ Amendment and Response.
- ☒ An additional fee is not required.
- ☐ The additional fee is calculated as shown below:

					SMALL ENTITY		LARGE ENTITY	
	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE	ADDIT. FEE	RATE	ADDIT. FEE
TOTAL	40	MINUS	40=	0	x25	$0	x50	$ 0
INDEP.	5	MINUS	5=	0	x100	$0	x200	$ 0
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIMS					+180	$	+360	$ 0
				TOTAL ADDITIONAL FEE		$0		$ 0

☒ The Commissioner is hereby authorized to charge any additional fees required under 37 CFR § 1.16, or to credit any overpayment to Deposit Account No. 11-0980. A duplicate of this sheet is enclosed.

I hereby certify that this correspondence is being facsimile transmitted to Mail Stop Amendment, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450, Attn: Examiner Jackson, GAU 2131, Facsimile No. (571) 273-8300 on January 11, 2007.

W. Scott Petty, Reg. No. 35,

KING & SPALDING LLP
34th Floor
1180 Peachtree Street, N.E.
Atlanta, Georgia 30309-3521
Telephone: 404.572.4600

By: _____
W. Scott Petty, Reg. No. 35,6

4395294 v1

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1162

DUPLICATE

PATENTS

RECEIVED
CENTRAL FAX CENTER
JAN 11 2007

IN THE U.S. PATENT AND TRADEMARK OFFICE

In re application of:

Thomas James Satterlee et al.

Docket No. **05456.105038**

Serial No.: **10/336,299**

Art Unit **2131**

Filed: **January 3, 2003**

For: **System and Method for the Managed Security Control of Processes of a Computer**

Examiner: **Jackson, Jenise E.**

Confirmation No.: **7850**

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

January 11, 2007

Sir:

Transmitted herewith are papers in the above-identified application.

- ☒ Amendment and Response.
- ☒ An additional fee is not required.
- ☐ The additional fee is calculated as shown below:

					SMALL ENTITY		LARGE ENTITY	
	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE	ADDIT. FEE	RATE	ADDIT. FEE
TOTAL	40	MINUS	40=	0	x25	$0	x50	$ 0
INDEP.	5	MINUS	5=	0	x100	$0	x200	$ 0
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIMS					+180	$	+360	$ 0
				TOTAL ADDITIONAL FEE		$0		$ 0

☒ The Commissioner is hereby authorized to charge any additional fees required under 37 CFR § 1.16, or to credit any overpayment to Deposit Account No. 11-0980. A duplicate of this sheet is enclosed.

I hereby certify that this correspondence is being facsimile transmitted to Mail Stop Amendment, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450, Attn: Examiner Jackson, GAU 2131, Facsimile No. (571) 273-8300 on January 11, 2007.

W. Scott Petty, Reg. No. 35,608

KING & SPALDING LLP
34th Floor
1180 Peachtree Street, N.E.
Atlanta, Georgia 30309-3521
Telephone: 404.572.4600

By: W. Scott Petty, Reg. No. 35,608

4395294 v1

DUPLICATE

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1163

RECEIVED
CENTRAL FAX CENTER

JAN 11 2007

Patents

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: **Thomas James Satterlee et al.**	
Serial No.: **10/336,299**	Art Unit: **2131**
Filing Date: **January 3, 2003**	Examiner: **Jackson, Jenise E.**
Title: **System and Method for the Managed Security Control of Processes of a Computer**	Confirmation No.: **7850** Attorney Docket No.: **05456.105038**

RESPONSE TO OFFICIAL ACTION

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

Sir:

Responsive to the Official Action mailed October 11, 2006, please consider the following remarks.

[This section is intentionally left blank.]

I hereby certify that this correspondence is being facsimile transmitted to Mail Stop Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, Attention: J. Jackson, GAU 2131, Facsimile No. 571-273-8300, on January 11, 2007.

/W. Scott Petty, Reg. No. 35,645/

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1164

RECEIVED
CENTRAL FAX CENTER

JAN 11 2007

REMARKS

The Applicants and the undersigned thank Examiner Jackson for a careful review of the present application. Consideration of this application is respectfully requested in view of the following remarks, which are responsive to the Official Action mailed October 11, 2006.

Claims 1-40 remain pending in the present application, with Claims 1, 13, 21, 26, and 36 being independent claims. The Examiner has rejected Claims 1-40.

I. The recitations of Claims 26-35 are disclosed within the specification

The Examiner issued rejections for Claims 26-35 under 35 U.S.C. § 112 asserting that the specification does not disclose and describe the recitations of:

"determining whether the file is restricted";

"taking remedial action to protect the computing device";

"if the file is not restricted, permitting the call to open the file";

"restricting"; and

"remedial actions".

The Applicants offer the following remarks to traverse the pending rejections.

Recitation of "determining whether the file is restricted"

Lines 24-27 on Page 16 of the specification disclose, "Any files that have restricted access, as determined by the user or network administrator in setup process 300, will be identified in the file lock data in database 210. The file protection monitor 135 consults the file lock data in step 720 to determine whether the subject file has been restricted." The Applicants respectfully submit that the section of the specification cited above provides the requisite disclosure and description.

Recitation of "taking remedial action to protect the computing device"

Lines 29-30 on Page 16 of the specification disclose, "However, if the file does appear in the file lock data 220 in step 735, the file protection monitor can limit access to the file." Furthermore, the follow paragraph of the specification gives other examples of remedial actions

PAGE 5/10 * RCVD AT 1/11/2007 3:22:05 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-5/18 * DNIS:2738300 * CSID:404 572 5134 * DURATION (mm-ss):03-14

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1165

for protecting the computing device. The Applicants respectfully submit that the sections of the specification referenced above provide the requisite disclosure and description.

Recitation of "if the file is not restricted, permitting the call to open the file"

Lines 27-29 on Page 16 of the specification disclose, "If the file is not restricted, the file protection monitor 135 permits the system call hook 175 to proceed with opening the file." The Applicants respectfully submit that the section of the specification cited above provides the requisite disclosure and description.

Recitation of "restricting"

Lines 10-13 on Page 12 of the specification disclose, "The file lock data 220 represents files selected by the user or network administrator that are to be restricted from access by programs running on the computing device 103." The Applicants respectfully submit that the section of the specification cited above provides the requisite disclosure and description.

Recitation of "remedial actions"

Lines 29-30 on Page 16 of the specification disclose, "However, if the file does appear in the file lock data 220 in step 735, the file protection monitor can limit access to the file." Furthermore, the follow paragraph of the specification gives other examples of remedial actions for protecting the computing device. The Applicants respectfully submit that the sections of the specification referenced above provide the requisite disclosure and description.

In view of the foregoing discussion of the specification's disclosure of recitations addressed in Claims 26-35, the Applicants courteously request the Examiner to withdrawal the pending rejections of these claims under 35 U.S.C. §112.

II. **Independent Claims 1, 13, 21, 26, and 36 are patentable over *Mateev***

The Examiner issued rejections for Claims 1-40 under 35 U.S.C. § 102(e) asserting that Published U.S. Patent Application Number 2003/0101381 to Mateev et al. (hereinafter

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1166

"*Mateev*") anticipates those claims. The Applicants offer the following remarks to traverse the pending rejections.

Claims 1 and 21

Claims 1 and 21 are independent claims defining methods of implementing security for a computing device. The Applicants respectfully submit that *Mateev* does not anticipate Claims 1 and 21, as the claims recite at least one feature that is not disclosed, not taught, and not suggested by *Mateev*. Moreover, Claims 1 and 21 each recite a combination of features that *Mateev* does not disclose.

For example, *Mateev* does not disclose, teach, or suggest identifying an allowed program that is permitted to execute on the computing device with the requirements of Claims 1 and 21. In contrast to identifying an allowed program, *Mateev* (pg. 9, 0082-0083) teaches the identification of code fragments that are not permitted to execute on some computer hardware by virtue of matching a known signature associated with a virus.

Rather than an identification based on a list of programs that are allowed as required by Claims 1 and 21, *Mateev's* identification is based on matching against a list of non-permitted code fragments. This difference is significant because malicious code may still fail to match one of the non-permitted patterns and pass undetected, while a program that is positively identified as allowed may have been specifically identified as trustworthy or non-malicious.

Further, *Mateev* does not teach making determinations on entire programs, but in contrast discusses identification of code fragments that are on the size scale of cache memory entries. Determinations made on cache entry scale fragments versus determinations made on entire programs may involve significant efficiency and cost impacts as many more, smaller elements are examined.

In view of the preceding discussion of distinctions between *Mateev* and Claims 1 and 21, the Applicants respectfully request withdrawal of the pending rejection of both Claims 1 and 21.

Claim 13

Claim 13 is an independent claim defining a method for implementing security on a computing device. The method of Claim 13 requires a step of monitoring the new program

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1167

while it loads and executes in connection with the computing device if the new program is not validated. The Applicants respectfully submit that *Mateev* does not anticipate Claim 13, as the claim recites at least one feature that is not disclosed, not taught, and not suggested by *Mateev*. Moreover, Claim 13 recites a combination of features that *Mateev* does not disclose.

Mateev teaches away from execution with a computing device as required by Claim 13, by making use of an abstraction layer, or virtualization layer, called the DELI (dynamic execution layer interface). The "DELI automatically takes control of an executing program in a manner in which the executing program is unaware that it is not executing directly on computer hardware". (See *Mateev*, pg. 5, 0053) In contract, Claim 13 requires a step of executing the program in connection with the computing device and monitoring the operations of the program. In Claim 13, the monitoring may afford protection from malicious code without preventing the execution of the program on the computing device. This is in contrast to *Mateev's* isolation of the code from the hardware by an abstraction layer, or virtualization layer.

Further, *Mateev* does not disclose, teach, or suggest any operation that is conditioned on if the new program is validated. As discussed in relationship to Claim 1, Mateev addresses the negative preclusion of code fragment execution and not the positive validation of entire programs. In contrast to identifying an allowed program, *Mateev* (pg. 9, 0082-0083) teaches the identification of code fragments that are not permitted to execute on some computer hardware by virtue of matching a known signature associated with a virus.

In view of the preceding discussion of distinctions between Claim 13 and *Mateev*, the Applicants respectfully request withdrawal of the pending rejection of Claim 13.

Claim 26

Claim 26 is an independent claim defining a method of performing security for a computing device during an execution phase. The Applicants respectfully submit that *Mateev* does not anticipate Claim 26, as the claim recites at least one feature that is not disclosed, not taught, and not suggested by *Mateev*. Moreover, Claim 26 recites a combination of features that *Mateev* does not disclose.

Mateev does not disclose, teach, or suggest restricting files with the requirements of Claim 26. Nor does *Mateev* disclose, teach, or suggest intercepting calls to open a file with the

PAGE 8/10 * RCVD AT 1/11/2007 3:22:05 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-5/18 * DNIS:2738300 * CSID:404 572 5134 * DURATION (mm-ss):03-14

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1168

requirements of Claim 26. Claim 26 recites a method for allowing certain files to be blocked, or restricted, from being accessed by potentially malicious programs. As such, Claim 26 can involve restricting access to data files, program files, or other files. Lines 10-13 on Page 12 of the specification clarifies, "The file lock data 220 represents files selected by the user or network administrator that are to be restricted from access by programs running on the computing device 103." In contrast to blocking certain files from being accessed by programs, *Mateev* (pg. 11, 0099) teaches the monitoring of the execution of code and makes no mention of maintaining a list of restricted files, or blocking code from accessing such restricted files. Further, *Mateev* (pg. 11, 0099) does not address the opening of files by the potentially malicious code.

In view of the preceding discussion of distinctions between *Mateev* and Claim 26, the Applicants respectfully request withdrawal of the pending rejection of Claim 26.

Claim 36

Claim 36 is an independent claim defining a system for managing security of a computing device. The Applicants respectfully submit that *Mateev* does not anticipate Claim 36, as the claim recites at least one feature that is not disclosed, not taught, and not suggested by *Mateev*. Moreover, Claim 36 recites a combination of features that *Mateev* does not disclose.

Mateev does not disclose, teach, or suggest a module for determining whether the program is valid as required by Claim 36. In contrast to determining whether the program is valid, *Mateev* (pg. 9, 0083-0085) teaches the screening for infected code to determine that the code is not permitted to execute by virtue of matching a known signature associated with a virus.

Rather than an identification based on programs that are valid, as required by Claim 36, *Mateev's* identification is based on not matching a list of code patterns that are not valid. This difference is significant because malicious code may still fail to match one of the non-valid patterns and pass undetected, while a program that is positively identified as allowed may have been specifically identified as trustworthy or non-malicious.

In view of the preceding discussion of distinctions between *Mateev* and Claim 36, the Applicants respectfully request withdrawal of the pending rejection of Claim 36.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1169

RECEIVED
CENTRAL FAX CENTER

JAN 11 2007

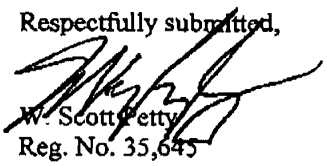
III. **Dependent Claims 2-12, 14-20, 22-25, 27-35, and 37-40 are patentable over the cited reference**

Each of dependent Claims 2-12, 14-20, 22-25, 27-35, and 37-40 incorporates the recitations of the respective claim or claims from which it depends. In view of the above-described distinctions between the reference cited by the Examiner and independent Claims 1, 13, 21, 26, and 36, the Applicants submit that dependent Claims 2-12, 14-20, 22-25, 27-35, and 37-40 are patentable over the reference that the Examiner has cited. Accordingly, the Applicants also respectfully request withdrawal of the Examiner's pending rejections of all the dependent claims. Furthermore, each of those dependent claims recites features and combinations of features further defining the present invention over the cited art. Therefore, the Applicants request separate and individual consideration of each dependent claim.

CONCLUSION

The foregoing is submitted as a full and complete response to the Official Action mailed October 11, 2006. The Applicants thank Examiner Jackson for consideration of the remarks presented by this response. The Applicants have shown that the pending claims are allowable and allowance of the claims is respectfully requested. It is believed that this response places the application in condition for allowance. Such action is courteously requested. If there are any issues that can be resolved with an Examiner's Amendment or a telephone conference, a telephone call to the undersigned at 404.572.2888 is respectfully requested.

Respectfully submitted,



W. Scott Petty
Reg. No. 35,645

King & Spalding LLP
34th Floor
1180 Peachtree Street, N.E.
Atlanta, Georgia 30309
404.572.4600

PAGE 10/10 * RCVD AT 1/11/2007 3:22:05 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-5/18 * DNIS:2738300 * CSID:404 572 5134 * DURATION (mm-ss):03-14

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1170

WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Wednesday, March 28, 2007

Hide?	Set Name	Query	Hit Count
	DB=PGPB,USPT,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR		
☐	L29	Internet near Security near systems.as. and (execut?)near4(program)	6
☐	L26	William near Frank near Hackenberger.in.	2
☐	L25	Thomas near James near Satterlee.in.	2
☐	L24	713/167.ccls.	364
☐	L23	713/188.ccls. and (pre-approv? or approv? or allow?)near5(program or executable or software or executable or signature or code)	70
☐	L22	713/187.ccls. and (pre-approv? or approv? or allow?)near5(program or executable or software or executable or signature or code)	73
☐	L21	713/182.ccls. and (pre-approv? or approv? or allow?)near5(program or executable or software or executable or signature or code)	182
☐	L20	713/165.ccls.	615
☐	L19	713/164.ccls.	399
☐	L18	713/161.ccls. and (pre-approv? or approv? or allow?)near5(program or executable or software or executable or signature or code)	39
☐	L17	726/26.ccls.	1183
☐	L16	726/24.ccls.	443
☐	L15	726/22.ccls.	792
☐	L14	726/14.ccls. and (pre-approv? or approv? or allow?)near5(program or executable or software or executable or signature or code)	44
☐	L13	726/2.ccls. and (pre-approv? or approv? or allow?)near5(program or executable or software or executable or signature or code)	71
☐	L12	726/1.ccls. and (pre-approv? or approv? or allow?)near5(program or executable or software or executable or signature or code)	116
☐	L11	L10 and L1	1277
☐	L10	(new)with(program or code or software)near4(chang? or modif? or alter?)	5027
☐	L9	(checksum or MD-5 or message near4 digest or digital near3 signature or digital near3 certificate) and L5	118
☐	L8	(compar? or authenticat? or authoriz?)with(new)with(program or software or code)near6(allow? or approv?)with(program or software or code)	13
☐	L7	(suspend?)with(execut? or run?)with(program or software or code)	60
☐	L6	(identif? or recogn?)with(allow? or approv?)near5(program or software or executable)	689
☐	L5	(authenticat? or validat? or permit? or authoriz? or verif?)near5(new)with (program or executable or software)	565

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1171

☐	L4	(new)near5(program or executable or software or executables or signature or program)	76524
☐	L3	L1 and L2	4892
☐	L2	(permit? or validat? or authenticat? or authoriz?)near5(program or executable or software or executables or signature or signatures)	20286
☐	L1	(pre-approv? or allow?)near5(program or executable or software or executable or signature or signatures)	63646

END OF SEARCH HISTORY

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1172

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.
10/336,299	01/03/2003	Thomas James Satterlee	05456.105038	7850

20786 7590 04/02/2007

KING & SPALDING LLP
1180 PEACHTREE STREET
ATLANTA, GA 30309-3521

EXAMINER	
JACKSON, JENISE E	
ART UNIT	PAPER NUMBER
2131	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	04/02/2007	PAPER

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

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1173

Office Action Summary	Application No.	Applicant(s)	
	10/336,299	SATTERLEE ET AL.	
	Examiner	Art Unit	
	Jenise E. Jackson	2131	

-- 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 *11 January 2007*.

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) *1-40* 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) *1-40* 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: _____.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1174

DETAILED ACTION

Claim Rejections - 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

2. Claims 1-40 are rejected under 35 U.S.C. 102(e) as being anticipated by Mateev et al(2003/0101381).

3. As per claim 1, Mateev et al. discloses computer-implemented method for implementing security for a computing device(pg. 4, 0039, 0040), identifying an allowed program that is permitted to execute on the computing device(see pg. 9, 0082-0083); receiving a signal that a new program is going to be executed on the computing device(see pg. 9, 0082-0083, 0087); suspending the execution of the new program on the computing device; determining whether the new program is the same as the allowed program(pg. 9 0083); if the new program is the same as the allowed program, permitting the new program to execute on the computing device; and if the new program is not the same as the allowed program, monitoring the new program while allowing it to execute on the computing device(see pg. 9, 0086-0087).

4. As per claim 2, Mateev discloses wherein the step of receiving a signal that a new program is going to be executed is performed at the operating system kernel of the computing device(pg. 5 0054, pg. 9 0083).

5. As per claim 3, Mateev discloses wherein the step of monitoring the new program while allowing it to execute is performed at the operating system kernel of the computing device(pg. 5, 0054, pg. 9 0083-0084).

6. As per claim 4, Mateev discloses wherein the step of determining whether the new program is the same as the allowed program(pg. 9, 0082-0083, 0087) computing a checksum for the new program, and comparing the new program's checksum to a checksum for the allowed program(pg. 12 0106).

7. As per claim 5, Mateev discloses wherein the step of determining whether the new program is the same as the allowed program comprises determining whether the new program has been modified (pg. 9 0082-0083, 0087).

8. As per claim 6, Mateev discloses wherein the step of monitoring the new program while allowing it to execute includes controlling the files the program attempts to access during execution of the new program on the computing device(pg. 9 0082-0083, 0087).

9. As per claim 7, Mateev discloses wherein the step of monitoring the new program while allowing it to execute includes controlling the registry settings the new program attempts to access during execution of the new program on the computing device(pg. 7 0066, 0068).

10. As per claim 8, Mateev discloses wherein the step of monitoring the new program while allowing it to execute includes controlling the network activity of the new program(pg. 0082-0083, 0087).

11. As per claim 9, Mateev discloses wherein the step of monitoring the new program while allowing it to execute includes determining whether a user has previously approved operation of the new program(pg. 9, 0083, pg. 11 0103).

12. As per claim 10, Mateev discloses the step of terminating the execution of the new program if the program performs suspicious activities(pg. 9 0084-0085).

13. As per claim 11, Mateev discloses the step of taking remedial measures to protect the security of other computing device components if the new program is not the same as the allowed program(pg. 9 0082-0083).

14. As per claim 12, Mateev discloses a computer-readable medium having computer-executable instructions for performing the steps recited in claim 1(pg. 4 0039).

15. As per claim 13, Mateev discloses a computer-implemented method for implementing security for a computing device(pg. 4, 0039, 0040), interrupting the loading of a new program for operation with the computing device; validating the new program; if the new program is validated, permitting the new program to continue loading and to execute in connection with the computing device; if the new program is not validated, monitoring the new program while it loads and executes in connection with the computing device(see pg. 9, 0083, 0086-0087).

16. As per claim 14, Mateev discloses wherein the step of interrupting the loading of a new program comprises: intercepting a signal from the computing device's operating system that the new program is loading, and suspending the loading of the new program(pg. 9 0083).

17. As per claim 15, Mateev discloses wherein the step of monitoring the new program includes intercepting a signal from the computing device's operating system(pg. 0082-0083, 0087).

18. As per claim 16, Mateev discloses wherein the step of validating the new program includes determining whether the new program corresponds with an approved program(pg. 9, 0083, 0086-0087).

19. As per claim 17, Mateev discloses wherein the step of validating the new program includes comparing a checksum for the new program with a previously determined checksum(pg. 12 0106).

20. As per claim 18, Mateev discloses wherein the step of validating the new program comprises analyzing characteristics of the new program(pg. 9 0084-0085).

21. As per claim 19, Mateev discloses wherein the step of monitoring the new program includes controlling the files the new program attempts to access during execution of the new program(pg. 9 0083).

22. As per claim 20, Mateev discloses a computer-readable medium having computer-executable instructions for performing the steps recited in claim 13(pg. 4 0039).

23. As per claim 21, Mateev discloses a computer-implemented method for performing security for a computing device during a pre-execution phase(pg. 9 0083) identifying an allowed program that is permitted to execute with the computing device; receiving a signal that a new program is being loaded for execution with the computing device; suspending the loading of the new program; comparing the new program to the allowed program; and determining whether the new program is valid(pg. 9 0083-0085).

24. As per claim 22, Mateev discloses wherein the step of receiving a signal includes receiving a signal from the computing device's operating system kernel(pg. 5 0054, pg. 9 0083).

25. As per claim 23, Mateev discloses wherein the step of comparing the new program to the allowed program includes comparing a checksum of the new program to a checksum of the allowed program(pg. 12 0106).

26. As per claim 24, Mateev discloses wherein the step of determining whether the new

program is valid includes determining whether a checksum of the new program corresponds to a checksum of the allowed program(pg. 12 0106).

27. As per claim 25, Mateev discloses a computer-readable medium having computer-executable instructions for performing the steps recited in claim 21(pg. 4 0039).

28. As per claim 26, Mateev discloses a computer-implemented method for performing security for a computing device during an execution phase(see pg. 9, 0082-0083), intercepting a call to open a file from the computing device's operating system; determining whether the file is restricted; if the file is restricted, taking remedial action to protect the computing device; and if the file is not restricted, permitting the call to open the file(pg. 11 0099).

29. As per claim 27, Mateev discloses the step of receiving an instruction to monitor a call from an executing program(pg. 11 0099, 0103).

30. As per claim 28, Mateev discloses the step of defining files that are restricted from access(pg. 11, 0105).

31. As per claim 29, Mateev discloses, wherein the step of defining files that are restricted includes designating a file that cannot be accessed in response to the call(pg. 11, 0103).

32. As per claim 30, Mateev discloses wherein the step of defining files that are restricted includes designating a file that cannot be modified in response to the call(pg. 9, 0091).

33. As per claim 31, Mateev discloses the step of alerting a security management module of the presence of a malicious program(pg. 11, 0103-0104).

34. As per claim 32, Mateev discloses wherein the step of taking remedial action includes terminating the call(pg. 10, 0094).

35. As per claim 33, Mateev discloses wherein the step of taking remedial action includes

alerting another security module of the existence of the call(pg. 11, 0104-0105).

36. As per claim 34, Mateev discloses wherein the step of taking remedial action includes notifying the user of the existence of the call(pg. 10, 0094).

37. As per claim 35, Mateev discloses a computer-readable medium having computer-executable instructions for performing the steps recited in claim 26(pg. 4 0039).

38. As per claim 36, Mateev discloses a system for managing security of a computing device a pre-execution module operable for receiving notice from the computing device's operating system that a program is being loaded onto the computing device(pg. 9 0083); a validation module coupled to the pre-execution monitor operable for determining whether the program is valid; a detection module coupled to the pre-execution monitor operable for intercepting a trigger from the computing device's operating system(pg. 9 0083-0085); and an execution module coupled to the detection module and operable for monitoring the program in response to the trigger intercepted by the detection module(see pg. 9, 0086-0087).

39. As per claim 37, Mateev discloses wherein the pre-execution module is further operable for suspending loading of the program onto the computing device(pg. 9 0083-0085).

40. As per claim 38, Mateev discloses wherein the pre-execution module is further operable for retrieving validation data(pg. 11 0103-0104).

41. As per claim 39, Mateev discloses wherein the execution module is further operable for deciding whether to terminate the trigger intercepted by the detection module(pg. 0105-0106).

42. As per claim 40, Mateev discloses wherein the execution module is further operable for retrieving authorization data and deciding how to respond to the trigger intercepted by the detection module(pg. 12 0107-0108).

Response to Applicant

43. Claims 26-35 were previously rejected under 112 1st and 2nd. The 112 rejection has been withdrawn. Applicant arguments are persuasive, in which the Applicant provides details and cites as to how claim limitations are provided in the specification.

44. Claims 1 and 21, the Applicant argues that Mateev does not disclose or suggest, identifying an allowed program that is permitted to execute on the computing device. The Examiner disagrees with the Applicant. Mateev discloses identifying infected code within a program[0039]. Mateev discloses a virus detection manager includes logic configured to intercept program instructions, and logic configured to determine if the program instructions are to be executed, and logic configured to identify if the program instructions are infected[0039]. The Examiner asserts that if a virus is not found in the program that the program is allowed to be executed.

45. Claim 13, the Applicant argues that Mateev does not disclose monitoring the new program while it loads and executes in connection with the computing device if the new program is not validated. The Examiner disagrees with the Applicant. Mateev discloses monitoring application code as it enters the DELI to determine if the code has been previously processed by the computing device; and when it is the case that the application code has not been previously processed, analyzing the program code to determine if infected code is present[0041].

46. Claim 26, the Applicant argues that Mateev does not disclose restricting files. The Examiner disagrees with the Applicant. Mateev discloses the code behavior checker is programmed to look for program activity indicative of common virus attacks. All programs

designated for execution by the computer system can be intercepted and processed[0085]. The Examiner asserts that these infected files are restricted files.

47. Also, as per claim 26, the Applicant argues that Mateev does not disclose intercepting a call to open a file. The Examiner disagrees with the Applicant. Mateev discloses deli start command is delivered by the Api that invokes the deli[0099]. Mateev discloses that the deli can be used to facilitate virus checking of program binaries[0100]. Mateev discloses the deli intercepting and executing application binaries[0101].

48. As per claim 36, the Applicant argues that Mateev does not disclose or suggest a module for determining whether the program is valid. The Examiner disagrees with the Applicant. Mateev discloses the virus detection manager may be programmed to read signature from a virus signature database that contains known signatures in viruses. When the program does not contain a match of the present signature a negative response is given. Once the signature comparison branch for identifying infected code has completed a flag may be set to indicate that the test is completed. A second flag may be set to indicate that the code fragment is clear of known virus signatures[0105]. The Examiner asserts that the program that is clear and does not contain a virus is a valid program.

Final Action

49. 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.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jenise E. Jackson whose telephone number is (571) 272-3791. The examiner can normally be reached on M-Th (6:00 a.m. - 3:30 p.m.) alternate Friday's.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ayaz Sheikh can be reached on (571) 272-3795. 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.

March 28, 2007

CHRISTOPHER REVAK
PRIMARY EXAMINER

Index of Claims



Application/Control No.	Applicant(s)/Patent under Reexamination
10/336,299	SATTERLEE ET AL.
Examiner	**Art Unit**
Jenise E. Jackson	2131

√	Rejected	–	(Through numeral) Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

Claim		Date
Final	Original	3/28/07
	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	
	32	
	33	
	34	
	35	
	36	
	37	
	38	
	39	
	40	√
	41	
	42	
	43	
	44	
	45	
	46	
	47	
	48	
	49	
	50	

Claim		Date
Final	Original	
	51	
	52	
	53	
	54	
	55	
	56	
	57	
	58	
	59	
	60	
	61	
	62	
	63	
	64	
	65	
	66	
	67	
	68	
	69	
	70	
	71	
	72	
	73	
	74	
	75	
	76	
	77	
	78	
	79	
	80	
	81	
	82	
	83	
	84	
	85	
	86	
	87	
	88	
	89	
	90	
	91	
	92	
	93	
	94	
	95	
	96	
	97	
	98	
	99	
	100	

Claim		Date
Final	Original	
	101	
	102	
	103	
	104	
	105	
	106	
	107	
	108	
	109	
	110	
	111	
	112	
	113	
	114	
	115	
	116	
	117	
	118	
	119	
	120	
	121	
	122	
	123	
	124	
	125	
	126	
	127	
	128	
	129	
	130	
	131	
	132	
	133	
	134	
	135	
	136	
	137	
	138	
	139	
	140	
	141	
	142	
	143	
	144	
	145	
	146	
	147	
	148	
	149	
	150	

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1185

Search Notes	Application/Control No.	Applicant(s)/Patent under Reexamination	
	10/336,299	SATTERLEE ET AL.	
	Examiner	Art Unit	
	Jenise E. Jackson	2131	

SEARCHED			
Class	Subclass	Date	Examiner
726	22	3/28/2007	JJ
726	24	3/28/2007	JJ
726	26	3/28/2007	JJ
713	164	3/28/2007	JJ
713	165	3/28/2007	JJ
713	167	3/28/2007	JJ

INTERFERENCE SEARCHED			
Class	Subclass	Date	Examiner

SEARCH NOTES (INCLUDING SEARCH STRATEGY)		
	DATE	EXMR
Updated west text search, see printout	3/28/2007	JJ
Updated west class search with text, see printout	3/28/2007	JJ
Updated inventor and assignee search, no double patenting, see printout	3/26/2007	JJ

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1186

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	PATENT NUMBER	GROUP ART UNIT	FILE WRAPPER LOCATION
10/336,299		2131	26M1

Correspondence Address / Fee Address Change

The following fields have been set to Customer Number 69151 on 04/03/2007

- Correspondence Address

The address of record for Customer Number 69151 is:
KING & SPALDING, LLP
INTELLECTUAL PROPERTY DEPT. - PATENTS
1180 PEACHTREE STREET, N.E.
ATLANTA,GA 30309-3521

RECEIVED
CENTRAL FAX CENTER
JUL 02 2007

KING & SPALDING, LLP

1180 Peachtree Street
Atlanta, Georgia 30309-3521
Telephone: 404/572-4600
Facsimile: 404/572-5100
www.kslaw.com

FAX TRANSMITTAL SHEET

July 2, 2007

TO:	Examiner Jenise E. Jackson GAU 2131 U.S. Serial No. 10/336,299		**Company:**	U.S. Patent and Trademark Office
	Fax #:	571-273-8300	**City/State:**	Alexandria, VA 22313
		Mail Stop AF		
FROM:	Robert T. Neufeld	5595	**Our Ref. #:**	05456.105038

NUMBER OF PAGES (including transmittal sheet): 3

CONFIDENTIALITY NOTICE

THE INFORMATION CONTAINED IN THIS FACSIMILE MESSAGE IS PRIVILEGED AND CONFIDENTIAL INFORMATION INTENDED FOR THE USE OF THE ADDRESSEE LISTED ABOVE. IF YOU ARE NEITHER THE INTENDED RECIPIENT NOR THE EMPLOYEE OR AGENT RESPONSIBLE FOR DELIVERING THIS MESSAGE TO THE INTENDED RECIPIENT, YOU ARE HEREBY NOTIFIED THAT ANY DISCLOSURE, COPYING, DISTRIBUTION OR THE TAKING OF ANY ACTION IN RELIANCE ON THE CONTENTS OF THIS TELECOPIED INFORMATION IS STRICTLY PROHIBITED. IF YOU HAVE RECEIVED THIS TELECOPY IN ERROR, PLEASE IMMEDIATELY NOTIFY US BY TELEPHONE TO ARRANGE FOR RETURN OF THE ORIGINAL DOCUMENTS TO US.

If transmission problems occur or you are not the intended recipient, please call 404.572.2459 immediately. Thank you.

Notes/Comments:

Documents Submitted Via Facsimile:

Applicant:	Thomas James Satterlee et al.
Serial No.:	10/336,299
Filed:	January 3, 2003
For:	System and Method for the Managed Security Control of Processes of a Computer
Papers Faxed:	Notice of Appeal (in duplicate) (2 pgs. total) ✓RTN

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1188

RECEIVED
CENTRAL FAX CENTER
JUL 02 2007

PATENTS

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: **Thomas James Satterlee et al.**	Atty Docket: **05456.105038**
Application No.: **10/336,299**	Art Unit: **2131**
Filing Date: **January 3, 2003**	Examiner: **Jackson, Jenise E.**
Title: **System and Method for the Managed Security Control of Processes of a Computer**	Confirmation No.: **7850**

NOTICE OF APPEAL

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

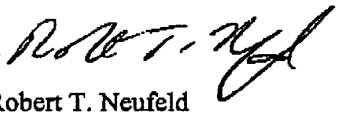
Sir:

Applicants hereby appeal to the Board of Patent Appeals and Interferences from the last decision of the Examiner dated April 2, 2007.

Please charge Deposit Account No. 11-0980 in the amount of $500.00 for the appeal fee required under 37 C.F.R. § 41.20(b)(1).

The Commissioner is authorized to charge any additional fee required for this Notice of Appeal, or to credit any overpayment, to Deposit Account No. 11-0980. A duplicate of this paper is enclosed.

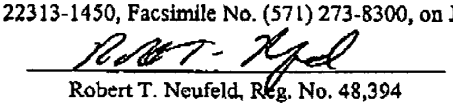
Respectfully submitted,



KING & SPALDING LLP
1180 Peachtree Street, N.E., 34th Floor
Atlanta, Georgia 30309-3521
(404) 572-4600

Robert T. Neufeld
Attorney for Applicants
Reg. No. 48,394

I hereby certify that this correspondence is being facsimile transmitted to: Mail Stop AF, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450, Facsimile No. (571) 273-8300, on July 2, 2007.



Robert T. Neufeld, Reg. No. 48,394

07/03/2007 TL0111 00000045 110980 10336299
01 FC:1491 500.00 DA

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1189

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	PATENT NUMBER	GROUP ART UNIT	FILE WRAPPER LOCATION
10/336,299		2131	26M1

Correspondence Address / Fee Address Change

The following fields have been set to Customer Number 20786 on 07/31/2007

- Correspondence Address
- Maintenance Fee Address

The address of record for Customer Number 20786 is:
KING & SPALDING LLP
1180 PEACHTREE STREET
ATLANTA,GA 30309-3521

PATENTS

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	)	
	)	Attorney Docket No. **05456.105038**
Thomas James Satterlee et al.	)	
	)	Group Art Unit: **2131**
Application No. **10/336,299**	)	
	)	Confirmation No.: **7850**
Filed: **January 3, 2003**	)	
	)	Examiner: **Jackson, Jenise E.**
For: **System and Method for the Managed Security Control of Processes of a Computer**	)	

APPEAL BRIEF

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

September 4, 2007

Sir:

Further to the Notice of Appeal filed on July 2, 2007 in the above referenced application, Appellants hereby submit this brief under 37 C.F.R. § 1.191 to appeal the Examiner's rejection of this application, as reported in the Final Office Action mailed on April 2, 2007.

Please charge $500.00 for the fee for filing a brief in support of the appeal required under 37 C.F.R. § 41.20(b)(2) to Deposit Account No. 09-0457.

The Commissioner is authorized to charge any additional fee required for this Notice of Appeal, or to credit any overpayment, to Deposit Account No. 09-0457.

Respectfully submitted,

/Robert T. Neufeld/

Robert T. Neufeld
Reg. No. 48,394

I hereby certify that this correspondence is being electronically transmitted to: Mail Stop Appeal Brief-Patents, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450, on September 4, 2007.

/James M. Hannon/
James M. Hannon, Reg. No. 48,565

Table of Contents

Real Party in Interest 3
Related Appeals and Interferences 4
Status of Claims 5
Status of Amendments 6
Summary of Claimed Subject Matter 7
Grounds of Rejection to be Reviewed on Appeal 9
Argument 10
Conclusion 18
APPENDIX 1 - Claims Appendix 19
APPENDIX 2 - Evidence Appendix 27
APPENDIX 3 - Related Proceedings Appendix 28

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1192

Real Party in Interest

The real party in interest is IBM Internet Security Services, formerly operating as Internet Security Systems, Inc., the assignee of record. IBM Internet Security Services is a wholly owned subsidiary of International Business Machines (IBM), Inc.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1193

Related Appeals and Interferences

None.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1194

Status of Claims

Claims 1-40 stand finally rejected and are on appeal.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1195

Status of Amendments

No amendments have been filed subsequent to the final rejection mailed on April 2, 2007.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1196

Summary of Claimed Subject Matter

In general, the invention disclosed by the present application defines a new system and process for managing and controlling the execution of software programs within a computing device to protect the computing device from malicious activities. A protector system can implement a two-step process to ensure that software programs do not perform malicious activities which may damage the computing device or other computing resources to which the device is coupled. In the first phase, the protector system can determine whether a software program has been previously approved and validate that the software program has not been altered. If the software program cannot be validated, the protector system can enter the second phase and detect and observe executing activities at the kernel level of the operating system so that suspicious actions can be anticipated and addressed before they are able to do harm to the computing device.

Independent Claims 1, 13, 21, and 26 are directed to methods for implementing security for a computing device. The method of Claim 1 includes the following steps: (1) identifying an allowed program that is permitted to execute on the computing device (page 11, lines 30 to 31; page 12, lines 18 to 22); (2) receiving a signal that a new program is going to be executed on the computing device (page 5, lines 14 to 16; page 13, lines 26 to 27); (3) suspending the execution of the new program on the computing device (page 14, lines 7 to 8; page 6, lines 1 to 3); (4) determining whether the new program is the same as the allowed program (page 5, lines 16 to 17; page 14, lines 12 to 15); (5) if the new program is the same as the allowed program, permitting the new program to execute on the computing device (page 5, lines 17 to 19; page 14, lines 15 to 17); and (6) if the new program is not the same as the allowed program, monitoring the new program while allowing it to execute on the computing device (page 5, lines 21 to 23; page 14 lines 17 to 20).

The method of Claim 13 includes the following steps: (1) interrupting the loading of a new program for operation with the computing device (page 14, lines 7 to 8; page 6, lines 1 to 3); (2) validating the new program (page 5, lines 16 to 17; page 14, lines 12 to 15); (3) if the new program is validated, permitting the new program to continue loading and to execute in connection with the computing device (page 5, lines 17 to 19; page 14, lines 15 to 17; and (4) if the new program is not validated, monitoring the new program while it loads and executes in connection with the computing device (page 5, lines 21 to 23; page 14 lines 17 to 20).

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1197

The method of Claim 21 includes the following steps: (1) identifying an allowed program that is permitted to execute with the computing device (page 11, lines 30 to 31; page 12, lines 18 to 22); (2) receiving a signal that a new program is being loaded for execution with the computing device (page 5, lines 14 to 16; page 13, lines 26 to 27); (3) suspending the loading of the new program (page 14, lines 7 to 8; page 6, lines 1 to 3); (4) comparing the new program to the allowed program (page 5, lines 16 to 17; page 14, lines 12 to 15); and (5) determining whether the new program is valid (page 5, lines 17 to 19; page 14, lines 15 to 17).

The method of Claim 26 includes the following steps: (1) intercepting a call to open a file from the computing device's operating system (page 16, lines 21 to 23); (2) determining whether the file is restricted (page 16, lines 26 to 27); (3) if the file is restricted, taking remedial action to protect the computing device (page 16, lines 29 to 30); and (4) if the file is not restricted, permitting the call to open the file (page 16, lines 27 to 29).

Independent Claim 36 is directed to a system for managing security of a computing device. The system of Claim 36 includes the following elements: (1) a pre-execution module operable for receiving notice from the computing device's operating system that a program is being loaded onto the computing device (page 5, lines 14 to 16; page 10, lines 27 to 30); (2) a validation module coupled to the pre-execution monitor operable for determining whether the program is valid (page 5, lines 16 to 17; page 12, lines 29 to 30); (3) a detection module coupled to the pre-execution monitor operable for intercepting a trigger from the computing device's operating system (page 5, lines 7 to 9; page 10, lines 22 to 23); and (4) an execution module coupled to the detection module and operable for monitoring the program in response to the trigger intercepted by the detection module (page 11, lines 23 to 26).

[This section is intentionally left blank.]

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1198

Grounds of Rejection to be Reviewed on Appeal

The following issue is presented on appeal:

(1) Whether, under 35 U.S.C. § 102(e), Claims 1-40 are anticipated by U.S. Publication No. 2003/0101381 ("*Mateev*").

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1199

Argument

Rejection of claims under 35 U.S.C. § 102(e)

The Legal Standard for 35 U.S.C. § 102(e)

An invention must be new at the time of discovery by an original inventor to be patentable. See Chisum, PATENTS, Vol. 1, § 3.01, p. 3-3. Section 101 of Title 35, U.S.C. requires that a patentable invention must be "new." The meaning of "new" is defined by specific conditions set forth in Section 102. Specifically, § 102, provides:

> A person shall be entitled to a patent unless . . . (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for the purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Under Section 102(e), the issue is strict novelty of the invention (i.e., anticipation). Thus, prior art references are examined to determine whether any one reference discloses the entire claimed invention within its "four corners." To anticipate the claimed invention, each and every element of the claimed invention must be disclosed in the reference. As mentioned, the standard for lack of novelty (i.e., anticipation) is one of strict identity. To anticipate a claim for a patent, a single prior source must contain all its essential elements. See Hybritech Inc. v. Monoclonal Antibodies, Inc., 802 F.2d 1367, 1379 (Fed. Cir. 1986) ("It is axiomatic that for prior art to anticipate under § 102 it has to meet every element of the claimed invention, and that such a determination is one of fact.

A traditional way of looking at the law of anticipation invalidity is as follows: "That which infringes, if later, would anticipate if earlier." Knapp v. Morss, 14 S.Ct. 81, 84 (1893). "That which will infringe, if later, will anticipate, if earlier. Thus a claim fails to meet the novelty requirement if it covers or reads on a product or process found in a single source in the prior art." Lewmar Marine, Inc. v. Barient, Inc., 827 F.2d 744, 747 (Fed. Cir. 1987) "The standard for lack of novelty, that is, for 'anticipation,' is one of strict identity. To anticipate a

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1200

claim for a patent, a single prior source must contain all its essential elements." See Chisum, § 3.02[1], p. 3-14. "The law of anticipation does not require that the reference 'teach' what the subject patent teaches. Assuming that a reference is properly 'prior art,' it is only necessary that the claims under attack, as construed by the court, 'read on' something disclosed in the reference, i.e., all limitations of the claim are found in the reference, or 'fully met' by it." Kalman v. Kimberly-Clark Corp., 713 F.2d 760, 771 (Fed. Cir. 1983).

Analysis

The *Mateev* reference disclosed by Examiner for the rejection of independent Claims 1, 13, 21, 26, and 36 under 35 U.S.C. § 102(e) fails to disclose recitations that are present in the independent claims.

Independent Claims 1 and 21

On pages 2 and 5 of the Final Office Action mailed on 4/2/07, the Examiner directs the Applicants' attention to page 9, lines 0082 to 0083 to teach the claimed feature of "identifying an allowed program that is permitted to execute on the computing device." However, this particular recitation from *Mateev* merely provides the identification of code fragments that are not permitted to execute on some computer hardware by virtue of matching a known signature associated with a virus. Rather than an identification based on a list of programs that are explicitly allowed as required by Claims 1 and 21, *Mateev's* identification is based on matching against a list of non-permitted code fragments. This difference is significant because malicious code may still fail to match one of the non-permitted patterns and pass undetected, while a program that is positively identified as allowed may have been specifically identified as trustworthy or non-malicious. Additionally *Mateev* does not provide any allowed program (presumably in a list or database of allowed programs) as required by Claims 1 and 21. On page 8 of the Final Office Action mailed on 4/2/07, the Examiner asserts "that if a virus is not found in the program that the program is allowed to be executed." However, merely testing against known virus signatures as a negative filtering process is entirely different from the positive identification of an entire program as one explicitly identified as an allowed program.

Further, *Mateev* does not teach making determinations on entire programs, but in contrast

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1201

discusses identification of code fragments that are on the size scale of cache memory entries. Determinations made on cache entry scale fragments versus determinations made on entire programs may involve significant efficiency and cost impacts as many more, smaller elements are examined.

In view of the preceding discussion of distinctions between *Mateev* and Claims 1 and 21, the Applicants respectfully request withdrawal of the pending rejection of both Claims 1 and 21.

Independent Claim 13

On page 4 of the Final Office Action mailed on 4/2/07, the Examiner directs the Applicants' attention to page 9, lines 0083, and 0096 to 0087 to teach the claimed features of "if the new program is not validated, monitoring the new program while it loads and executes in connection with the computing device" as required by Claim13. However, *Mateev* teaches away from execution with a computing device as required by Claim 13, by making use of an abstraction layer, or virtualization layer, called the DELI (dynamic execution layer interface). The "DELI automatically takes control of an executing program in a manner in which the executing program is unaware that it is not executing directly on computer hardware". (See *Mateev*, pg. 5, 0053) In contract, Claim 13 requires a step of executing the program in connection with the computing device and monitoring the operations of the program. In Claim 13, the monitoring may afford protection from malicious code without preventing the execution of the program on the computing device. This is in contrast to *Mateev's* isolation of the code from the hardware by an abstraction layer, or virtualization layer.

Further, *Mateev* does not disclose, teach, or suggest any operation that is conditioned on if the new program is validated. As discussed in relationship to Claim 1, Mateev addresses the negative preclusion of code fragment execution and not the positive validation of entire programs. In contrast to identifying an allowed program, *Mateev* (pg. 9, 0082-0083) teaches the identification of code fragments that are not permitted to execute on some computer hardware by virtue of matching a known signature associated with a virus.

In view of the preceding discussion of distinctions between Claim 13 and *Mateev*, the Applicants respectfully request withdrawal of the pending rejection of Claim 13.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1202

Independent Claim 26

On page 6 of the Final Office Action mailed on 4/2/07, the Examiner directs the Applicants' attention to page 11, lines 0099, and 0096 to 0087 to teach the claimed features of "intercepting a call to open a file from the computing device's operating system; determining whether the file is restricted; if the file is restricted, taking remedial action to protect the computing device; and if the file is not restricted, permitting the call to open the file." However, *Mateev* does not disclose, teach, or suggest restricting files with the requirements of Claim 26. Nor does *Mateev* disclose, teach, or suggest intercepting calls to open a file with the requirements of Claim 26. Claim 26 recites a method for allowing certain files to be blocked, or restricted, from being accessed by potentially malicious programs. As such, Claim 26 can involve restricting access to data files, program files, or other files. Lines 10-13 on Page 12 of the specification clarifies, "The file lock data 220 represents files selected by the user or network administrator that are to be restricted from access by programs running on the computing device 103." In contrast to blocking certain files from being accessed by programs, *Mateev* (pg. 11, 0099) teaches the monitoring of the execution of code and makes no mention of maintaining a list of restricted files, or blocking code from accessing such restricted files. Further, *Mateev* (pg. 11, 0099) does not address the opening of files by the potentially malicious code.

On pages 8 and 9 of the Final Office Action mailed on 4/2/07, the Examiner asserts "that these infected filed are restricted files." Again, the infected files of *Mateev* are the program files that are attempting to execute and have been identified as malicious code or as viruses, while in contrast the restricted file of Claim 26 is a local file, such as a data file, that is restricted from access so as to protect it from the malicious code that is attempting to execute. *Mateev* addresses viruses, and thus restricting execution, Claim 26 addresses local files, and thus restricted access, not only execution.

In view of the preceding discussion of distinctions between *Mateev* and Claim 26, the Applicants respectfully request withdrawal of the pending rejection of Claim 26.

Independent Claim 36

On page 7 of the Final Office Action mailed on 4/2/07, the Examiner directs the Applicants' attention to page 9, lines 0083 to 0087 to teach the claimed features of determining whether the program is valid. However, *Mateev* teaches the screening for infected code to

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1203

determine that the code is not permitted to execute by virtue of matching a known signature associated with a virus.

Rather than an identification based on programs that are valid, as required by Claim 36, *Mateev's* identification is based on not matching a list of code patterns that are not valid. This difference is significant because malicious code may still fail to match one of the non-valid patterns and pass undetected, while a program that is positively identified as allowed may have been specifically identified as trustworthy or non-malicious.

In view of the preceding discussion of distinctions between *Mateev* and Claim 36, the Applicants respectfully request withdrawal of the pending rejection of Claim 36.

Dependent Claims 2-12, 14-20, 22-25, 27-35, and 37-40

Applicants respectfully submit that the above-identified dependent claims are allowable because the independent claims from which they depend are patentable over the cited documents. Applicants also respectfully submit that the recitations of these dependent claims are of patentable significance.

In view of the foregoing, Applicants respectfully request that the Examiner withdraw the pending rejections of dependent Claims 2-12, 14-20, 22-25, 27-35, and 37-40.

Rejection of claims under 35 U.S.C. § 103(a)

While the Examiner did not base the rejection on 35 U.S.C. § 103, Applicants nonetheless submit that the prior art also does not render the pending claims obvious.

The Legal Standard for 35 U.S.C. § 103

The U.S. Patent and Trademark Office has the burden under 35 U.S.C. § 103 to establish a *prima facie* case of obviousness. *In re Warner et al.*, 379 F.2d 1011, 154 U.S.P.Q. 173, 177 (C.C.P.A. 1967), *In re Fine,* 837 F.2d 1071, 1074, 5 U.S.P.Q.2d 1596, 1598-99 (Fed. Cir. 1988). To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1204

success must both be found in the prior art and not based on applicants' disclosure. *In re Vaeck, 947 F.2d 488, 20 U.S.P.Q.2d 1438 (Fed. Cir. 1991).* The references cited by the Examiner do not meet all three criteria.

While the "teaching, suggestion, motivation test" should not be applied rigidly, it nonetheless provides insight into the obviousness inquiry. KSR Int'l v. Teleflex Inc., 82 U.S.P.Q.2d 1385, 1391, 1396 (2007). The prior art must provide one of ordinary skill in the art with the motivation to make the proposed modification needed to arrive at the claimed invention. *In re Geiger*, 815 F.2d 686, 2 U.S.P.Q.2d 1276 (Fed. Cir. 1987); *in re Lalu and Foulletier*, 747 F.2d 703, 705, 223 U.S.P.Q. 1257, 1258 (Fed. Cir. 1984). Claims for an invention are not *prima facie* obvious if the primary references do not suggest all elements of the claimed invention and the prior art does not suggest the modifications that would bring the primary references into conformity with the application claims. *In re Fritch*, 23 U.S.P.Q.2d, 1780 (Fed. Cir. 1992). *In re Laskowski*, 871 F.2d 115 (Fed. Cir. 1989). This is not possible when the claimed invention achieves more than what any or all of the prior art reference allegedly suggest, expressly or by reasonable implication.

It is clear that to establish a rejection under 35 U.S.C. § 103 the cited references must (1) recite each element of the claims, (2) provide one of skill in the art with the motivation to combine the cited reference as applications have done and (3) provide one of ordinary skill in the art with a reasonable expectation of success. The references cited by the Examiner clearly do not meet all three criteria and the current rejection of the claims-in-issue lack proper support.

Analysis

Independent Claims 1 and 21

The *Mateev* reference disclosed by the Examiner fails to disclose or suggest the claimed feature of "identifying an allowed program that is permitted to execute on the computing device." In contrast, *Mateev* merely recites the identification of code fragments that are not permitted to execute on some computer hardware by virtue of matching a known signature associated with a virus. Rather than an identification based on a list of programs that are explicitly allowed as required by Claims 1 and 21, *Mateev's* identification is based on matching against a list of non-permitted code fragments. This difference is significant because malicious

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1205

code may still fail to match one of the non-permitted patterns and pass undetected, while a program that is positively identified as allowed may have been specifically identified as trustworthy or non-malicious. Additionally *Mateev* does not recite allowed programs (presumably in a list or database of allowed programs) as required by Claims 1 and 21. Merely testing against known virus signatures as a negative filtering process is entirely different from the positive identification of an entire program as one explicitly identified as an allowed program.

Further, *Mateev* does not teach making determinations on entire programs, but in contrast discusses identification of code fragments that are on the size scale of cache memory entries. Determinations made on cache entry scale fragments versus determinations made on entire programs may involve significant efficiency and cost impacts as many more, smaller elements are examined.

Independent Claim 13

The *Mateev* reference disclosed by the Examiner fails to disclose or suggest the claimed feature of "if the new program is not validated, monitoring the new program while it loads and executes in connection with the computing device" as required by Claim13. Instead, *Mateev* teaches away from execution with a computing device as required by Claim 13, by making use of an abstraction layer, or virtualization layer, called the DELI (dynamic execution layer interface). The "DELI automatically takes control of an executing program in a manner in which the executing program is unaware that it is not executing directly on computer hardware". (See *Mateev*, pg. 5, 0053) In contrast, Claim 13 requires a step of executing the program in connection with the computing device and monitoring the operations of the program. In Claim 13, the monitoring may afford protection from malicious code without preventing the execution of the program on the computing device. This is in contrast to *Mateev's* isolation of the code from the hardware by an abstraction layer, or virtualization layer.

Further, *Mateev* does not disclose, teach, or suggest any operation that is conditioned on if the new program is validated. As discussed in relationship to Claim 1, Mateev addresses the negative preclusion of code fragment execution and not the positive validation of entire programs. In contrast to identifying an allowed program, *Mateev* (pg. 9, 0082-0083) teaches the identification of code fragments that are not permitted to execute on some computer hardware by virtue of matching a known signature associated with a virus.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1206

Independent Claim 26

The *Mateev* reference disclosed by the Examiner fails to disclose or suggest the claimed feature of "intercepting a call to open a file from the computing device's operating system; determining whether the file is restricted; if the file is restricted, taking remedial action to protect the computing device; and if the file is not restricted, permitting the call to open the file." The *Mateev* reference does not disclose, teach, or suggest restricting files with the requirements of Claim 26. Nor does *Mateev* disclose, teach, or suggest intercepting calls to open a file with the requirements of Claim 26. Claim 26 recites a method for allowing certain files to be blocked, or restricted, from being accessed by potentially malicious programs. As such, Claim 26 can involve restricting access to data files, program files, or other files. Lines 10-13 on Page 12 of the specification clarifies, "The file lock data 220 represents files selected by the user or network administrator that are to be restricted from access by programs running on the computing device 103." In contrast to blocking certain files from being accessed by programs, *Mateev* (pg. 11, 0099) teaches the monitoring of the execution of code and makes no mention of maintaining a list of restricted files, or blocking code from accessing such restricted files. Further, *Mateev* (pg. 11, 0099) does not address the opening of files by the potentially malicious code.

Independent Claim 36

The *Mateev* reference disclosed by the Examiner fails to disclose or suggest the claimed feature of determining whether the program is valid. In contrast, *Mateev* teaches the screening for infected code to determine that the code is not permitted to execute by virtue of matching a known signature associated with a virus.

Rather than an identification based on programs that are valid, as required by Claim 36, *Mateev's* identification is based on not matching a list of code patterns that are not valid. This difference is significant because malicious code may still fail to match one of the non-valid patterns and pass undetected, while a program that is positively identified as allowed may have been specifically identified as trustworthy or non-malicious.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1207

Dependent Claims 2-12, 14-20, 22-25, 27-35, and 37-40

Applicants respectfully submit that the above-identified dependent claims are allowable because the independent claims from which they depend are patentable over the cited documents. Applicants also respectfully submit that the recitations of these dependent claims are of patentable significance.

Conclusion

In view of the arguments presented herein, Applicants respectfully request that the final rejection in this matter be vacated, and that this application be returned to the Examiner with instructions to enter a notice of allowance.

Respectfully submitted,

/Robert T. Neufeld/

Robert T. Neufeld
Reg. No. 48,394

KING & SPALDING LLP
1180 Peachtree Street
34th Floor
Atlanta, GA 30309
(404) 572-4600 (Telephone)
(404) 572-5134 (Facsimile)

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1208

APPENDIX 1

CLAIMS APPENDIX

1. (Original) A computer-implemented method for implementing security for a computing device, comprising the steps of:

identifying an allowed program that is permitted to execute on the computing device;

receiving a signal that a new program is going to be executed on the computing device;

suspending the execution of the new program on the computing device;

determining whether the new program is the same as the allowed program;

if the new program is the same as the allowed program, permitting the new program to execute on the computing device; and

if the new program is not the same as the allowed program, monitoring the new program while allowing it to execute on the computing device.

2. (Original) The method of Claim 1, wherein the step of receiving a signal that a new program is going to be executed is performed at the operating system kernel of the computing device.

3. (Original) The method of Claim 1, wherein the step of monitoring the new program while allowing it to execute is performed at the operating system kernel of the computing device.

4. (Original) The method of Claim1, wherein the step of determining whether the new program is the same as the allowed program comprises:

computing a checksum for the new program, and

comparing the new program's checksum to a checksum for the allowed program.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1209

5. (Original) The method of Claim 1, wherein the step of determining whether the new program is the same as the allowed program comprises determining whether the new program has been modified.

6. (Original) The method of Claim 1, wherein the step of monitoring the new program while allowing it to execute comprises controlling the files the program attempts to access during execution of the new program on the computing device.

7. (Original) The method of Claim 1, wherein the step of monitoring the new program while allowing it to execute comprises controlling the registry settings the new program attempts to access during execution of the new program on the computing device.

8. (Original) The method of Claim 1, wherein the step of monitoring the new program while allowing it to execute comprises controlling the network activity of the new program.

9. (Original) The method of Claim 1, wherein the step of monitoring the new program while allowing it to execute comprises determining whether a user has previously approved operation of the new program.

10. (Original) The method of Claim 1, further comprising the step of terminating the execution of the new program if the program performs suspicious activities.

11. (Original) The method of Claim 1, further comprising the step of taking remedial measures to protect the security of other computing device components if the new program is not the same as the allowed program.

12. (Original) A computer-readable medium having computer-executable instructions for performing the steps recited in Claim 1.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1210

13. (Original) A computer-implemented method for implementing security for a computing device comprising the steps of:

interrupting the loading of a new program for operation with the computing device;

validating the new program;

if the new program is validated, permitting the new program to continue loading and to execute in connection with the computing device;

if the new program is not validated, monitoring the new program while it loads and executes in connection with the computing device.

14. (Original) The method of Claim 13, wherein the step of interrupting the loading of a new program comprises:

intercepting a signal from the computing device's operating system that the new program is loading, and

suspending the loading of the new program.

15. (Original) The method of Claim 13, wherein the step of monitoring the new program comprises intercepting a signal from the computing device's operating system.

16. (Original) The method of Claim 13, wherein the step of validating the new program comprises determining whether the new program corresponds with an approved program.

17. (Original) The method of Claim 13, wherein the step of validating the new program comprises comparing a checksum for the new program with a previously determined checksum.

18. (Original) The method of Claim 13, wherein the step of validating the new program comprises analyzing characteristics of the new program.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1211

19. (Original) The method of Claim 13, wherein the step of monitoring the new program comprises controlling the files the new program attempts to access during execution of the new program.

20. (Original) A computer-readable medium having computer-executable instructions for performing the steps recited in Claim 13.

[This section is intentionally left blank.]

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1212

21. (Original) A computer-implemented method for performing security for a computing device during a pre-execution phase comprising the steps of:

identifying an allowed program that is permitted to execute with the computing device;

receiving a signal that a new program is being loaded for execution with the computing device;

suspending the loading of the new program;

comparing the new program to the allowed program; and

determining whether the new program is valid.

22. (Original) The method of Claim 21, wherein the step of receiving a signal comprises receiving a signal from the computing device's operating system kernel.

23. (Original) The method of Claim 21, wherein the step of comparing the new program to the allowed program comprises comparing a checksum of the new program to a checksum of the allowed program.

24. (Original) The method of Claim 21, wherein the step of determining whether the new program is valid comprises determining whether a checksum of the new program corresponds to a checksum of the allowed program.

25. (Original) A computer-readable medium having computer-executable instructions for performing the steps recited in Claim 21.

[This section is intentionally left blank.]

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1213

26. (Original) A computer-implemented method for performing security for a computing device during an execution phase comprising the steps of:

intercepting a call to open a file from the computing device's operating system;

determining whether the file is restricted;

if the file is restricted, taking remedial action to protect the computing device; and

if the file is not restricted, permitting the call to open the file.

27. (Original) The method of Claim 26, further comprising the step of receiving an instruction to monitor a call from an executing program.

28. (Original) The method of Claim 26, further comprising the step of defining files that are restricted from access.

29. (Original) The method of Claim 28, wherein the step of defining files that are restricted comprises designating a file that cannot be accessed in response to the call.

30. (Original) The method of Claim 28, wherein the step of defining files that are restricted comprises designating a file that cannot be modified in response to the call.

31. (Original) The method of Claim 26, further comprising the step of alerting a security management module of the presence of a malicious program.

32. (Original) The method of Claim 26, wherein the step of taking remedial action comprises terminating the call.

33. (Original) The method of Claim 26, wherein the step of taking remedial action comprises alerting another security module of the existence of the call.

34. (Original) The method of Claim 26, wherein the step of taking remedial action comprises notifying the user of the existence of the call.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1214

35. (Original) A computer-readable medium having computer-executable instructions for performing the steps recited in Claim 26.

[This section is intentionally left blank.]

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1215

36. (Original) A system for managing security of a computing device comprising:

a pre-execution module operable for receiving notice from the computing device's operating system that a program is being loaded onto the computing device;

a validation module coupled to the pre-execution monitor operable for determining whether the program is valid;

a detection module coupled to the pre-execution monitor operable for intercepting a trigger from the computing device's operating system; and

an execution module coupled to the detection module and operable for monitoring the program in response to the trigger intercepted by the detection module.

37. (Original) The system of Claim 36, wherein the pre-execution module is further operable for suspending loading of the program onto the computing device.

38. (Original) The system of Claim 36, wherein the pre-execution module is further operable for retrieving validation data.

39. (Original) The system of Claim 36, wherein the execution module is further operable for deciding whether to terminate the trigger intercepted by the detection module.

40. (Original) The system of Claim 36, wherein the execution module is further operable for retrieving authorization data and deciding how to respond to the trigger intercepted by the detection module.

[This section is intentionally left blank.]

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1216

APPENDIX 2

EVIDENCE APPENDIX

None.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1217

APPENDIX 3

RELATED PROCEEDINGS APPENDIX

None.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1218

Electronic Patent Application Fee Transmittal

Application Number:	10336299
Filing Date:	03-Jan-2003
Title of Invention:	System and method for the managed security control of processes on a computer system
First Named Inventor/Applicant Name:	Thomas James Satterlee
Filer:	James Michael Hannon
Attorney Docket Number:	05456.105038
Filed as Large Entity	

Utility Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Filing a brief in support of an appeal	1402	1	500	500
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
			Total in USD ($)	**500**

Electronic Acknowledgement Receipt

EFS ID:	2154753
Application Number:	10336299
International Application Number:	
Confirmation Number:	7850
Title of Invention:	System and method for the managed security control of processes on a computer system
First Named Inventor/Applicant Name:	Thomas James Satterlee
Customer Number:	20786
Filer:	James Michael Hannon
Filer Authorized By:	
Attorney Docket Number:	05456.105038
Receipt Date:	04-SEP-2007
Filing Date:	03-JAN-2003
Time Stamp:	16:09:19
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment was successfully received in RAM	$500
RAM confirmation Number	1292
Deposit Account	090457
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows: Charge any Additional Fees required under 37 C.F.R. Section 1.16 and 1.17	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes) /Message Digest	Multi Part /.zip	Pages (if appl.)
1	Appeal Brief Filed	Appeal_Brief_IBM_105038_Sept_4_2007_28_pgs.pdf	160654 91adae19b5e56af121845717a2d099ce9a99fc8b	no	28
Warnings:					
Information:					
2	Fee Worksheet (PTO-06)	fee-info.pdf	8184 02882ffc9eafb679223fb1dc69fc89573f6b559e	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	168838		

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	PATENT NUMBER	GROUP ART UNIT	FILE WRAPPER LOCATION
10/336,299		2131	26M1

Correspondence Address / Fee Address Change

The following fields have been set to Customer Number 69151 on 09/05/2007

- Correspondence Address
- Maintenance Fee Address

The address of record for Customer Number 69151 is:
KING & SPALDING, LLP
INTELLECTUAL PROPERTY DEPT. - PATENTS
1180 PEACHTREE STREET, N.E.
ATLANTA,GA 30309-3521

WEST Search History

Hide Items | Restore | Clear | Cancel

updated search

DATE: Saturday, December 08, 2007

Hide?	Set Name	Query	Hit Count
	DB=PGPB,USPT,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR		
☐	L38	Internet near Security near Systems.as. and (execut?)near4(program)	9
☐	L37	William near Frank near Hackenberger.in.	3
☐	L36	Thomas near James near Satterlee.in.	3
☐	L34	713/167.ccls.	471
☐	L33	713/188.ccls.and (pre-approv? or approv? or allow?)near5(program or executable or software)	57
☐	L32	713/187.ccls.and (pre-approv? or approv? or allow?)near5(program or executable or software)	65
☐	L31	713/182.ccls.and (pre-approv? or approv? or allow?)near5(program or executable or software)	154
☐	L30	713/165.ccls.	754
☐	L29	713/164.ccls.	479
☐	L28	713/161.ccls.and (pre-approv? or approv? or allow?)near5(program or executable or software)	32
☐	L27	726/26.ccls.	1443
☐	L26	726/24.ccls.	592
☐	L25	726/22.ccls.	1006
☐	L24	726/14.ccls.and (pre-approv? or approv? or allow?)near5(program or executable or software)	40
☐	L23	726/2.ccls.and (pre-approv? or approv? or allow?)near5(program or executable or software)	76
☐	L22	726/1.ccls.and (pre-approv? or approv? or allow?)near5(program or executable or software)	106
☐	L21	(compar? or authenticat? or authoriz? or verif? or validat?)with(new)near5 (program or file or executable)near5(approv? or allow?)near5(program or file or executable)	3
☐	L20	L15 and L1	71
☐	L19	(compar? or validat?)near5(checksum or MD-5)with(new)near5(program or software or executable)	5
☐	L18	L17 and L11	880
☐	L17	(checksum or MD-5 or message near4 digestor digital nar3 signature or digital near3 certificate) and L15	1476
☐	L16	(identif? or recogn?)with(allow? or approv?)near5(program or software or executable) and L15	2

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1224

☐	L15	(authenticat$4 or validat$4 or permit$4 or authoriz$4 or verif$4)near5(new) with(program or executable or software)	2312
☐	L14	L12 and L13	3689
☐	L13	(new)near5(program or executable or software or executable or signature or program)	84595
☐	L12	(permit? or validat? or authenticat? or authoriz?)near5(program or executable or software or executable or signature) and L11	9489
☐	L11	(pre-approv? or allow$4 or approv$4)near5(program or executable or software or signature or file)	159172
☐	L10	(control$4 or manag$4)near5(system near4 settings or registry) and L9	762
☐	L9	(control$4 or manag$4)near4(file or software) and (monitor$4)near5(program or executable or software)	22418
☐	L8	(monitor$4)near5(kernel) and L1	42
☐	L7	(kernel)and L1	331
☐	L6	(check$3 or look$3)near5(list or database)near5(allow$4 or approv$4)near5 (program or executable or file)	88
☐	L5	L3 and L4	180
☐	L4	(suspend$3 or halt$3)near5(execution)near5(program or file or executable)	2354
☐	L3	(new)near5(program or executable or file) and (allowed or approv$4 or authoriz$4)near5(program or executable or file)	7072
☐	L2	(new)near5(program or executable or file)near5(compar$4 or validat$4 or authenticat$4)with(allow$4 or approv$4)near5(program or executable or file)	26
☐	L1	(allowed or approv$4 or authoriz$4)near5(program or executable or file)near5 (execut$4 or run$4)	2615

END OF SEARCH HISTORY

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1225

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.
10/336,299	01/03/2003	Thomas James Satterlee	05456.105038	7850

69151 7590 12/12/2007

KING & SPALDING, LLP
INTELLECTUAL PROPERTY DEPT. - PATENTS
1180 PEACHTREE STREET, N.E.
ATLANTA, GA 30309-3521

EXAMINER	
JACKSON, JENISE E	
ART UNIT	**PAPER NUMBER**
2131	

MAIL DATE	DELIVERY MODE
12/12/2007	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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1226

mN

Office Action Summary	Application No.	Applicant(s)	
	10/336,299	SATTERLEE ET AL.	
	Examiner	Art Unit	
	Jenise E. Jackson	2131	

-- 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 *04 September 2007*.

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) *1-40* 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) *1-40* 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: _____.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1227

DETAILED ACTION

1. In view of the Applicant's arguments filed on 9/4/07, PROSECUTION IS HEREBY REOPENED. A new art rejection has been applied as set forth below.

To avoid abandonment of the application, appellant must exercise one of the following two options:

(1) file a reply under 37 CFR 1.111 (if this Office action is non-final) or a reply under 37 CFR 1.113 (if this Office action is final); or,

(2) initiate a new appeal by filing a notice of appeal under 37 CFR 41.31 followed by an appeal brief under 37 CFR 41.37. The previously paid notice of appeal fee and appeal brief fee can be applied to the new appeal. If, however, the appeal fees set forth in 37 CFR 41.20 have been increased since they were previously paid, then appellant must pay the difference between the increased fees and the amount previously paid.

A Supervisory Patent Examiner (SPE) has approved of reopening prosecution by signing below:

GILBERTO BARRON JR
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

3. Claims 1-40 are rejected under 35 U.S.C. 102(e) as being anticipated by McGee et al(6,694,434)

4. As per claim 1, McGee et al. discloses identifying an allowed program that is permitted to execute on the computing device(see col. 4, lines 1-23); receiving a signal that a new program is going to be executed on the computing device(see col. 6, lines 39-42, 46-50); suspending the execution of the new program on the computing device; determining whether the new program is the same as the allowed program(col. 8, lines 11-40); if the new program is the same as the allowed program, permitting the new program to execute on the computing device(see col. 12, lines 52-63); and if the new program is not the same as the allowed program, monitoring the new program while allowing it to execute on the computing device(see col. 6, lines 19-50).

4. As per claim 2, McGee discloses wherein the step of receiving a signal that a new program is going to be executed is performed at the operating system kernel of the computing device(see col. 11, lines 7-12, col. 12, lines 52-63).

5. As per claim 3, McGee discloses wherein the step of monitoring the new program while allowing it to execute is performed at the operating system kernel of the computing device(see col. 6, lines 19-50, col. 11, lines 7-12).

6. As per claim 4, McGee discloses wherein the step of determining whether the new program is the same as the allowed program(see col. 12, lines 52-63), computing a checksum for the new program, and comparing the new program's checksum to a checksum for the allowed program(col. 4, lines 35-49).

7. As per claim 5, McGee discloses wherein the step of determining whether the new program is the same as the allowed program comprises determining whether the new program

has been modified (col. 9, lines 12-37).

8. As per claim 6, McGee discloses wherein the step of monitoring the new program while allowing it to execute includes controlling the files the program attempts to access during execution of the new program on the computing device(see col. 11, lines 3-36, col. 12, lines 19-63).

9. As per claim 7, McGee discloses wherein the step of monitoring the new program while allowing it to execute includes controlling the registry settings the new program attempts to access during execution of the new program on the computing device(col. 6, lines 51-67, col. 12, lines 52-63).

10. As per claim 8, McGee discloses wherein the step of monitoring the new program while allowing it to execute includes controlling the network activity of the new program(col. 6, lines 51-67, col. 8, lines 11-40).

11. As per claim 9, McGee discloses wherein the step of monitoring the new program while allowing it to execute includes determining whether a user has previously approved operation of the new program(see col. 12, lines 19-63).

12. As per claim 10, McGee discloses the step of terminating the execution of the new program if the program performs suspicious activities(see col. 4, lines 40-61).

13. As per claim 11, McGee discloses the step of taking remedial measures to protect the security of other computing device components if the new program is not the same as the allowed program(col. 8, lines 11-40).

14. As per claim 12, McGee discloses a computer-readable medium having computer-executable instructions for performing the steps recited in claim 1(see col. 10, lines 23-38).

15. As per claim 13, McGee discloses a computer-implemented method for implementing security for a computing device(see col. 10, lines 23-38), interrupting the loading of a new program for operation with the computing device; validating the new program; if the new program is validated, permitting the new program to continue loading and to execute in connection with the computing device; if the new program is not validated, monitoring the new program while it loads and executes in connection with the computing device(see col. 12, lines 19-63).

16. As per claim 14, McGee discloses wherein the step of interrupting the loading of a new program comprises: intercepting a signal from the computing device's operating system that the new program is loading, and suspending the loading of the new program(col. 8, lines 11-40).

17. As per claim 15, McGee discloses wherein the step of monitoring the new program includes intercepting a signal from the computing device's operating system((see col. 11, lines 7-12, col. 12, lines 52-63).

18. As per claim 16, McGee discloses wherein the step of validating the new program includes determining whether the new program corresponds with an approved program(col. 6, lines 19-50).

19. As per claim 17, McGee discloses wherein the step of validating the new program includes comparing a checksum for the new program with a previously determined checksum((col. 4, lines 35-49).

20. As per claim 18, McGee discloses wherein the step of validating the new program comprises analyzing characteristics of the new program(col. 8, lines 11-40).

21. As per claim 19, McGee discloses wherein the step of monitoring the new program

includes controlling the files the new program attempts to access during execution of the new program(see col. 11, lines 3-36, col. 12, lines 19-63).

22. As per claim 20, McGee discloses a computer-readable medium having computer-executable instructions for performing the steps recited in claim 13(see col. 10, lines 23-38).

23. As per claim 21, McGee discloses a computer-implemented method for performing security for a computing device during a pre-execution phase(col. 10, lines 23-38) identifying an allowed program that is permitted to execute with the computing device(see col. 4, lines 1-23); receiving a signal that a new program is being loaded for execution with the computing device(see col. 11, lines 7-12, col. 12, lines 52-63); suspending the loading of the new program; comparing the new program to the allowed program(see col. 8, lines 11-40); and determining whether the new program is valid(col. 12, lines 52-63).

24. As per claim 22, McGee discloses wherein the step of receiving a signal includes receiving a signal from the computing device's operating system kernel(col. 11, lines 7-12, col. 12, lines 52-63).

25. As per claim 23, McGee discloses wherein the step of comparing the new program to the allowed program includes comparing a checksum of the new program to a checksum of the allowed program(col. 4, lines 35-49).

26. As per claim 24, McGee discloses wherein the step of determining whether the new program is valid includes determining whether a checksum of the new program corresponds to a checksum of the allowed program(see col. 4, lines 35-49).

27. As per claim 25, McGee discloses a computer-readable medium having computer-executable instructions for performing the steps recited in claim 21(col. 10, lines 23-38).

28. As per claim 26, McGee discloses a computer-implemented method for performing security for a computing device during an execution phase(see col. 10, lines 23-38), intercepting a call to open a file from the computing device's operating system(see col. 11, lines 7-12); determining whether the file is restricted; if the file is restricted, taking remedial action to protect the computing device; and if the file is not restricted, permitting the call to open the file(see col. 8, lines 11-40).

29. As per claim 27, McGee discloses the step of receiving an instruction to monitor a call from an executing program(see col. 11, lines 7-12).

30. As per claim 28, McGee discloses the step of defining files that are restricted from access(see col. 11, lines 17-35).

31. As per claim 29, McGee discloses, wherein the step of defining files that are restricted includes designating a file that cannot be accessed in response to the call(see col. 11, lines 3-36).

32. As per claim 30, McGee discloses wherein the step of defining files that are restricted includes designating a file that cannot be modified in response to the call(see col. 11, lines 3-36, col. 13, lines 8-20).

33. As per claim 31, McGee discloses the step of alerting a security management module of the presence of a malicious program(see col. 13, lines 8-20).

34. As per claim 32, McGee discloses wherein the step of taking remedial action includes terminating the call(see col. 8, lines 11-40).

35. As per claim 33, McGee discloses wherein the step of taking remedial action includes alerting another security module of the existence of the call(see col. 8, lines 11-40, col. 13, lines 8-20).

36. As per claim 34, McGee discloses wherein the step of taking remedial action includes notifying the user of the existence of the call(see col. 8, lines 11-40, col. 11, lines 7-12).

37. As per claim 35, McGee discloses a computer-readable medium having computer-executable instructions for performing the steps recited in claim 26(see col. 10, lines 23-38).

38. As per claim 36, McGee discloses a system for managing security of a computing device a pre-execution module operable for receiving notice from the computing device's operating system that a program is being loaded onto the computing device; a validation module coupled to the pre-execution monitor operable for determining whether the program is valid(see col. 6, lines 3-19); a detection module coupled to the pre-execution monitor operable for intercepting a trigger from the computing device's operating system(see col. 11, lines 3-12, 38-67, col. 12, lines 1-51); and an execution module coupled to the detection module and operable for monitoring the program in response to the trigger intercepted by the detection module(see col. 12, lines 15-18).

39. As per claim 37, McGee discloses wherein the pre-execution module is further operable for suspending loading of the program onto the computing device(see col. 8, lines 11-40).

40. As per claim 38, McGee discloses wherein the pre-execution module is further operable for retrieving validation data(see col. 6, lines 3-19).

41. As per claim 39, McGee discloses wherein the execution module is further operable for deciding whether to terminate the trigger intercepted by the detection module(see col. 4, lines 40-61).

42. As per claim 40, McGee discloses wherein the execution module is further operable for retrieving authorization data and deciding how to respond to the trigger intercepted by the detection module(see col. 6, lines 3-19).

Response to Applicant

43. The Applicants arguments were persuasive and thus new art has been applied to the claims.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jenise E. Jackson whose telephone number is (571) 272-3791. The examiner can normally be reached on M-Th (6:00 a.m. - 3:30 p.m.) alternate Friday's.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ayaz Sheikh can be reached on (571) 272-3795. 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.

12/7/07

GILBERTO BARRON JR
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Notice of References Cited	Application/Control No. 10/336,299	Applicant(s)/Patent Under Reexamination SATTERLEE ET AL.	
	Examiner Jenise E. Jackson	Art Unit 2131	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-6,694,434	02-2004	McGee et al.	713/189
	B	US-6,334,213	12-2001	Li, Weijun	717/170
	C	US-			
	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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1236

Index of Claims

Application/Control No.	Applicant(s)/Patent under Reexamination
10/336,299	SATTERLEE ET AL.
Examiner	**Art Unit**
Jenise E. Jackson	2131

√	Rejected	–	(Through numeral) Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

Claim		Date
Final	Original	12/8/07
	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	
	32	
	33	
	34	
	35	
	36	
	37	
	38	
	39	↓
	40	√
	41	
	42	
	43	
	44	
	45	
	46	
	47	
	48	
	49	
	50	

Claim		Date
Final	Original	
	51	
	52	
	53	
	54	
	55	
	56	
	57	
	58	
	59	
	60	
	61	
	62	
	63	
	64	
	65	
	66	
	67	
	68	
	69	
	70	
	71	
	72	
	73	
	74	
	75	
	76	
	77	
	78	
	79	
	80	
	81	
	82	
	83	
	84	
	85	
	86	
	87	
	88	
	89	
	90	
	91	
	92	
	93	
	94	
	95	
	96	
	97	
	98	
	99	
	100	

Claim		Date
Final	Original	
	101	
	102	
	103	
	104	
	105	
	106	
	107	
	108	
	109	
	110	
	111	
	112	
	113	
	114	
	115	
	116	
	117	
	118	
	119	
	120	
	121	
	122	
	123	
	124	
	125	
	126	
	127	
	128	
	129	
	130	
	131	
	132	
	133	
	134	
	135	
	136	
	137	
	138	
	139	
	140	
	141	
	142	
	143	
	144	
	145	
	146	
	147	
	148	
	149	
	150	

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1237

Search Notes

Application/Control No.	Applicant(s)/Patent under Reexamination	
10/336,299	SATTERLEE ET AL.	
Examiner	**Art Unit**	
Jenise E. Jackson	2131	

SEARCHED			
Class	Subclass	Date	Examiner
726	22	12/8/2007	JJ
726	24	12/8/2007	JJ
726	26	12/8/2007	JJ
713	164	12/8/2007	JJ
713	165	12/8/2007	JJ
713	167	12/8/2007	JJ

INTERFERENCE SEARCHED			
Class	Subclass	Date	Examiner

SEARCH NOTES (INCLUDING SEARCH STRATEGY)		
	DATE	EXMR
Updated West text search, see printout	12/8/2007	JJ
Updated West class search with text, see printout	12/8/2007	JJ
Updated Inventor and Assignee search no double patenting, see printout	12/8/2007	JJ

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1238

RECEIVED
CENTRAL FAX CENTER
MAR 10 2008

KING & SPALDING

King & Spalding LLP
1180 Peachtree Street N.E.
Atlanta, Georgia 30309-3521
Telephone: 404/572-4600
Facsimile: 404/572-5100
www.kslaw.com

FACSIMILE

Important Notice: This facsimile transmission is intended to be delivered only to the named addressee, and may contain material that is confidential, proprietary or subject to legal protection or privilege. If it is received by anyone other than the named addressee, the recipient should immediately notify the sender at the address and telephone number set forth herein and obtain instructions as to the disposal of the transmitted material. In no event should such material be read or retained by anyone other than the named addressee, except by express authority of the sender or the named addressee.

DATE: March 10, 2008

Recipient	Company	City/State	Telephone #	Fax #
U.S. Application Serial No. 10/336,299	USPTO	Alexandria, VA		571.273.8300

FROM: Robert T. Neufeld 5595 **Our Ref. #:** 05456.105038

NUMBER OF PAGES (Including Cover Page): 4

MESSAGE:

Applicant:	Thomas James Satterlee et al.
Application No.	10/336,299
Filed:	January 3, 2003
For:	System and Method for the Managed Security Control of Processes of a Computer
Paper Submitted:	Fax Cover Sheet (1 pg.) Notice of Appeal and a copy (2 pg.) Credit Card Form - PTO-2038 (1 pg.)

RTN

PLEASE CHECK THAT ALL PAGES ARE RECEIVED. IN CASE OF PROBLEMS, PLEASE CALL **Robert T. Neufeld at 404.572.3505.**

ALL RETURN TELECOPY MESSAGES SHOULD BE SENT TO **404/572-5100**. THANK YOU.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1239

RECEIVED
CENTRAL FAX CENTER
MAR 10 2008

PATENTS

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: **Thomas James Satterlee et al.**	Atty Docket: **05456.105038**
Application No.: **10/336,299**	Art Unit: **2131**
Filing Date: **January 3, 2003**	Examiner: **Jackson, Jenise E.**
Title: **System and Method for the Managed Security Control of Processes of a Computer**	Confirmation No.: **7850**

NOTICE OF APPEAL

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

Sir:

Applicants hereby appeal to the Board of Patent Appeals and Interferences from the last decision of the Examiner mailed December 12, 2007.

Please apply the $500 Notice of Appeal fee previously paid with the notice submitted on July 2, 2007. Please charge the remaining fee of $10 to the credit card account indicated in the attached Form PTO-2038 credit card form.

The Commissioner is authorized to charge any additional fee required for this Notice of Appeal, or to credit any overpayment, to Deposit Account No. 11-0980. A duplicate of this paper is enclosed.

Respectfully submitted,

KING & SPALDING LLP
1180 Peachtree Street, N.E., 34th Floor
Atlanta, Georgia 30309-3521
(404) 572-4600

Robert T. Neufeld
Attorney for Applicants
Reg. No. 48,394

I hereby certify that this correspondence is being facsimile transmitted to: Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450, Facsimile No. (571) 273-8300, on March 10, 2008.

Robert T. Neufeld, Reg. No. 48,394.

Adjustment date: 03/12/2008 HVUONG1
07/03/2007 TLO111 00000045 110980 10336299
01 FC:1401 -500.00 CR

03/12/2008 HVUONG1 00000022 110980 10336299
01 FC:1401 500.00 DA 10.00 OP

PAGE 3/4 * RCVD AT 3/10/2008 5:01:27 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-4/2 * DNIS:2738300 * CSID:404 572 5134 * DURATION (mm-ss):01-40

KING & SPALDING

King & Spalding LLP
1180 Peachtree Street N.E.
Atlanta, Georgia 30309-3521
Telephone: 404/572-4600
Facsimile: 404/572-5100
www.kslaw.com

RECEIVED
CENTRAL FAX CENTER
MAY 09 2008

FACSIMILE

Important Notice: This facsimile transmission is intended to be delivered only to the named addressee, and may contain material that is confidential, proprietary or subject to legal protection or privilege. If it is received by anyone other than the named addressee, the recipient should immediately notify the sender at the address and telephone number set forth herein and obtain instructions as to the disposal of the transmitted material. In no event should such material be read or retained by anyone other than the named addressee, except by express authority of the sender or the named addressee.

DATE: May 9, 2008

Recipient	Company	City/State	Telephone #	Fax #
U.S. Application Serial No. 10/336,299	USPTO	Alexandria, VA		571.273.8300

FROM: Robert T. Neufeld 5595 **Our Ref. #:** 05456.105038

NUMBER OF PAGES (Including Cover Page): 26

MESSAGE:

Applicant:	Thomas James Satterlee et al.
Application No.	10/336,299
Filed:	January 3, 2003
For:	System and Method for the Managed Security Control of Processes of a Computer
Paper Submitted:	Fax Cover Sheet (1 pg.) Appeal Brief (23 pgs.) Fee Transmittal Sheet (original and copy 2 pgs.)

PTN

PLEASE CHECK THAT ALL PAGES ARE RECEIVED. IN CASE OF PROBLEMS, PLEASE CALL **Robert T. Neufeld at 404.572.3505.**

ALL RETURN TELECOPY MESSAGES SHOULD BE SENT TO **404/572-5100**. THANK YOU.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1241

PATENTS

IN THE U.S. PATENT AND TRADEMARK OFFICE

In re Application of:	**Thomas James Satterlee et al**	Attorney Docket No.:	**05456.105038**
		Art Unit:	**2131**
Serial No.:	**10/336,299**	Examiner:	**Jackson, Jenise E.**
Filing Date:	**January 3, 2003**	Confirmation No.:	**7850**
Title:	**System and Method for the Managed Security Control of Processes on a Computer**		

RECEIVED
CENTRAL FAX CENTER
MAY 09 2008

TRANSMITTAL

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Transmitted herewith are papers in the above-identified application.

- ☒ Appeal Brief.
- ☒ Faxsheet.
- ☐ An additional fee is not required.
- ☒ The additional fee is calculated as shown below:

	CLAIMS REMAINING ADDIT. AFTER AMENDMENT		HIGHEST PREVIOUSLY PAID FOR	PRESENT EXTRA	SMALL ENTITY RATE	SMALL ENTITY ADDIT. FEE	LARGE ENTITY RATE	LARGE ENTITY FEE
TOTAL	25	MINUS	40	0	x25	$ 0	x50	$
INDEP.	3	MINUS	5	0	x100	$ 0	x200	$
Remaining Appeal Brief Fee ($500.00 previously paid)								10.00
				TOTAL ADDITIONAL FEE		$ 0		$10.00

☒ The Commissioner is hereby authorized to charge any additional fees required under 37 CFR §1.16, or credit any overpayment to Account No. 09-0457. A duplicate copy of this sheet is provided.

I hereby certify that this correspondence is being facsimile transmitted to: Appeal Brief-Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450, Facsimile No. (571) 273-8300 on May 9, 2008.

Robert T. Neufeld, Reg. No. 48,394

KING & SPALDING LLP
1180 Peachtree Street
Atlanta, Georgia 30309
Telephone: **404.572.4600**

By: Robert T. Neufeld
Robert T. Neufeld, Reg. No. 48,394

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1242

PATENTS

IN THE U.S. PATENT AND TRADEMARK OFFICE

In re Application of:	**Thomas James Satterlee et al**	Attorney Docket No.:	**05456.105038**
		Art Unit:	**2131**
Serial No.:	**10/336,299**	Examiner:	**Jackson, Jenise E.**
Filing Date:	**January 3, 2003**	Confirmation No.:	**7850**
Title:	**System and Method for the Managed Security Control of Processes on a Computer**		

RECEIVED
CENTRAL FAX CENTER
MAY 09 2008

TRANSMITTAL

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Transmitted herewith are papers in the above-identified application.

- ☒ Appeal Brief.
- ☒ Faxsheet.
- ☐ An additional fee is not required.
- ☒ The additional fee is calculated as shown below:

					SMALL ENTITY		LARGE ENTITY	
	CLAIMS REMAINING ADDIT. AFTER AMENDMENT		HIGHEST PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE	ADDIT. FEE	RATE	FEE
TOTAL	25	MINUS	40	0	x25	$ 0	x50	$
INDEP.	3	MINUS	5	0	x100	$ 0	x200	$
Remaining Appeal Brief Fee ($500.00 previously paid)								10.00
				TOTAL ADDITIONAL FEE		$ 0		$10.00

☒ The Commissioner is hereby authorized to charge any additional fees required under 37 CFR §1.16, or credit any overpayment to Account No. 09-0457. A duplicate copy of this sheet is provided.

I hereby certify that this correspondence is being facsimile transmitted to: Appeal Brief-Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450, Facsimile No. (571) 273-8300 on May 9, 2008.

Robert T. Neufeld, Reg. No. 48,394

KING & SPALDING LLP
1180 Peachtree Street
Atlanta, Georgia 30309
Telephone: **404.572.4600**

By: Robert T. Neufeld, Reg. No. 48,394

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1243

RECEIVED
CENTRAL FAX CENTER
MAY 09 2008

PATENTS

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: **Thomas James Satterlee et al.**	Attorney Docket No. **05456.105038**
Application No. **10/336,299**	Group Art Unit: **2131**
Filed: **January 3, 2003**	Confirmation No.: **7850**
For: **System and Method for the Managed Security Control of Processes on a Computer**	Examiner: **Jackson, Jenise E.**

APPEAL BRIEF

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

May 9, 2008

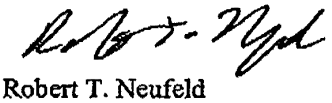
Sir:

Further to the Notice of Appeal filed on March 10, 2008 in the above referenced application, Appellants hereby submit this brief under 37 C.F.R. § 1.191 to appeal the Examiner's rejection of this application, as reported in the Office Action mailed on December 12, 2007.

Please apply the $500.00 appeal brief fee paid with the previous appeal brief submitted September 4, 2007. Please charge the remaining fee of $10 to Deposit Account No. 09-0457.

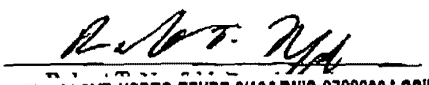
The Commissioner is authorized to charge any additional fee required for this Appeal Brief, or to credit any overpayment, to Deposit Account No. 09-0457.

Respectfully submitted,



Robert T. Neufeld
Reg. No. 48,394

I hereby certify that this correspondence is being facsimile transmitted to: Mail Stop Appeal Brief-Patents, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450, fax no. 571-273-8300, on May 9, 2008.



PAGE 4/26 * RCVD AT 5/9/2008 5:26:04 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-6/46 * DNIS:2738300 * CSID:404 572 5135 * DURATION (mm-ss):06-12

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1244

Table of Contents

Real Party in Interest 3

Related Appeals and Interferences 4

Status of Claims 5

Status of Amendments 6

Summary of Claimed Subject Matter 7

Grounds of Rejection to be Reviewed on Appeal 8

Argument 9

Conclusion 15

APPENDIX 1 16

APPENDIX 2 22

APPENDIX 3 23

PAGE 5/26 * RCVD AT 5/9/2008 5:26:04 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-6/46 * DNIS:2738300 * CSID:404 572 5135 * DURATION (mm-ss):06-12

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1245

RECEIVED
CENTRAL FAX CENTER
MAY 09 2008

Real Party in Interest

The real party in interest is IBM Internet Security Services, formerly operating as Internet Security Systems, Inc. IBM Internet Security Services is a wholly owned subsidiary of International Business Machines Corporation.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1246

Related Appeals and Interferences

None.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1247

Status of Claims

Claims 1-40 stand rejected in the most recent Office Action of December 12, 2007.

The rejection of Claims 1-20 and 36-40 is appealed.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1248

Status of Amendments

No amendments have been filed subsequent to the final rejection mailed on December 12, 2007.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1249

RECEIVED
CENTRAL FAX CENTER
MAY 09 2008

Summary of Claimed Subject Matter

The method of Claim 1 includes the following steps: (1) identifying an allowed program that is permitted to execute on the computing device (page 11, lines 30 to 31; page 12, lines 18 to 22); (2) receiving a signal that a new program is going to be executed on the computing device (page 5, lines 14 to 16; page 13, lines 26 to 27); (3) suspending the execution of the new program on the computing device (page 14, lines 7 to 8; page 6, lines 1 to 3); (4) determining whether the new program is the same as the allowed program (page 5, lines 16 to 17; page 14, lines 12 to 15); (5) if the new program is the same as the allowed program, permitting the new program to execute on the computing device (page 5, lines 17 to 19; page 14, lines 15 to 17); and (6) if the new program is not the same as the allowed program, monitoring the new program while allowing it to execute on the computing device (page 5, lines 21 to 23; page 14 lines 17 to 20).

The method of Claim 13 includes the following steps: (1) interrupting the loading of a new program for operation with the computing device (page 14, lines 7 to 8; page 6, lines 1 to 3); (2) validating the new program (page 5, lines 16 to 17; page 14, lines 12 to 15); (3) if the new program is validated, permitting the new program to continue loading and to execute in connection with the computing device (page 5, lines 17 to 19; page 14, lines 15 to 17; and (4) if the new program is not validated, monitoring the new program while it loads and executes in connection with the computing device (page 5, lines 21 to 23; page 14 lines 17 to 20).

Independent Claim 36 is directed to a system for managing security of a computing device. The system of Claim 36 includes the following elements: (1) a pre-execution module operable for receiving notice from the computing device's operating system that a program is being loaded onto the computing device (page 5, lines 14 to 16; page 10, lines 27 to 30); (2) a validation module coupled to the pre-execution monitor operable for determining whether the program is valid (page 5, lines 16 to 17; page 12, lines 29 to 30); (3) a detection module coupled to the pre-execution monitor operable for intercepting a trigger from the computing device's operating system (page 5, lines 7 to 9; page 10, lines 22 to 23); and (4) an execution module coupled to the detection module and operable for monitoring the program in response to the trigger intercepted by the detection module (page 11, lines 23 to 26).

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1250

RECEIVED
CENTRAL FAX CENTER
MAY 09 2008

Grounds of Rejection to be Reviewed on Appeal

The following issue is presented on appeal:

(1) Whether, under 35 U.S.C. § 102(e), Claims 1-20 and 36-40 are anticipated by U.S. Patent No. 6,694,434 ("*Magee*").

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1251

Argument

RECEIVED
CENTRAL FAX CENTER
MAY 09 2008

Rejection of claims under 35 U.S.C. § 102(e)

The Legal Standard for 35 U.S.C. § 102(e)

An invention must be new at the time of discovery by an original inventor to be patentable. See Chisum, PATENTS, Vol. 1, § 3.01, p. 3-3. Section 101 of Title 35, U.S.C. requires that a patentable invention must be "new." The meaning of "new" is defined by specific conditions set forth in Section 102. Specifically, § 102, provides:

> A person shall be entitled to a patent unless . . . (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for the purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Under Section 102(e), the issue is strict novelty of the invention (i.e., anticipation). Thus, prior art references are examined to determine whether any one reference discloses the entire claimed invention within its "four corners." To anticipate the claimed invention, each and every element of the claimed invention must be disclosed in the reference. As mentioned, the standard for lack of novelty (i.e., anticipation) is one of strict identity. To anticipate a claim for a patent, a single prior source must contain all its essential elements. See Hybritech Inc. v. Monoclonal Antibodies, Inc., 802 F.2d 1367, 1379 (Fed. Cir. 1986) ("It is axiomatic that for prior art to anticipate under § 102 it has to meet every element of the claimed invention, and that such a determination is one of fact.

A traditional way of looking at the law of anticipation invalidity is as follows: "That which infringes, if later, would anticipate if earlier." Knapp v. Morss, 14 S.Ct. 81, 84 (1893). "That which will infringe, if later, will anticipate, if earlier. Thus a claim fails to meet the novelty requirement if it covers or reads on a product or process found in a single source in the prior art." Lewmar Marine, Inc. v. Barient, Inc., 827 F.2d 744, 747 (Fed. Cir. 1987) "The standard for lack of novelty, that is, for 'anticipation,' is one of strict identity. To anticipate a

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1252

claim for a patent, a single prior source must contain all its essential elements." See Chisum, § 3.02[1], p. 3-14. "The law of anticipation does not require that the reference 'teach' what the subject patent teaches. Assuming that a reference is properly 'prior art,' it is only necessary that the claims under attack, as construed by the court, 'read on' something disclosed in the reference, i.e., all limitations of the claim are found in the reference, or 'fully met' by it." Kalman v. Kimberly-Clark Corp., 713 F.2d 760, 771 (Fed. Cir. 1983).

Analysis

The *Magee* reference disclosed by Examiner for the rejection of independent Claims 1, 13, and 36 under 35 U.S.C. § 102(e) fails to disclose recitations that are present in the independent claims.

Independent Claims 1 and 13

On page 3 of the Office Action mailed on 12/12/07, the Examiner directs the Applicants' attention to column 6, lines 19-50 of *Magee* to teach monitoring the new program while allowing it to execute on the computing device if the new program is different as required by Claim 1. Similarly, on page 5 of the same Office Action, the Examiner cites column 12, lines 19-63 of *Magee* to teach monitoring of the new program while allowing it to execute on the computing device if the new program is not validated as required by Claim 13.

However, Applicants respectfully submit that *Magee* fails to disclose monitoring of a program while it executes as required by Claims 1 and 13. The cited paragraph at lines 19-50 of column 6 describes an initial validation process for an application that is similar to the initial determining or validating steps recited in Claims 1 and 13. These initial determining or validating steps necessarily occur prior to the last step in each of Claims 1 and 13 where execution of the program is allowed to proceed but is monitored. The passages in the cited paragraph in column 6 of *Magee* refer to validating the application by comparing hash values and, if there is no match, requesting approval from the user. In contrast, both Claims 1 and 13 recite a process whereby if the new program is different or not validated, the new program will be allowed to execute, but will be monitored.

At page 5 of the Office Action, the Examiner also cites column 12, lines 19-63. However, similar to the passage at column 6, the paragraph at column 12 only describes an initial

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1253

comparison using hash values. The paragraph at column 12 explains "[e]xecution of the new version is allowed based on a comparison of the stored hash of the upgrade . . . to a computed hash . . . ." (*Magee*, col. 12, lines 52-54, emphasis added.) Applicants submit the cited paragraph at column 12 does not describe a process whereby after it is determined that a new program is not validated, allowing the new program to execute while monitoring the program.

Applicants submit the other portions of the *Magee* specification also fail to disclose monitoring a new program while it executes after determining the new program is different or not validated. For example, in column 11 *Magee* states "[t]his detection is preferably done on a continuous basis so that no executable file is allowed to execute without being analyzed." (Col. 11, lines 12-14.) "[I]f the hash value generated by the receiving processor does not match the has value on the approved hash list, the system prevents the executable file data from executing . . . ." (Col. 11, lines 54-57.) Applicants submit that *Magee* does not address the scenarios recited in Claims 1 and 13 where the new program, although not approved, is allowed to execute while being monitored.

In view of the preceding discussion of distinctions between Claims 1 and 13 and *Magee*, the Applicants respectfully request withdrawal of the pending rejection of Claims 1 and 13.

Independent Claim 36

On page 8 of the Office Action mailed on 12/12/07, the Examiner directs the Applicants' attention to column 12, lines 15-18 to teach the claimed features of an execution module coupled to a detection module and operable for monitoring a program in response to a trigger intercepted by the detection module. However, Applicants submit the cited passage of *Magee* merely describes individual receiving processors that perform their own comparison for an application by comparing a hash value the individual processor generates against a hash value at a central trusted authority.

Applicants submit that the cited passage from *Magee* does not describe an execution module that monitors an executing program. In contrast, the cited passage from *Magee* describes a processor performing a verification step using a generated hash value that is compared to a trusted hash value. Furthermore, *Magee* does not disclose the execution module performing monitoring of a program in response to a trigger intercepted by a detection module.

In view of the preceding discussion of distinctions between *Magee* and Claim 36, the

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1254

Applicants respectfully request withdrawal of the pending rejection of Claim 36.

Dependent Claims 2-12, 14-20, and 37-40

Applicants respectfully submit that the above-identified dependent claims are allowable because the independent claims from which they depend are patentable over the cited documents. Applicants also respectfully submit that the recitations of these dependent claims are of patentable significance.

In view of the foregoing, Applicants respectfully request that the Examiner withdraw the pending rejections of dependent Claims 2-12, 14-20, and 37-40.

Rejection of claims under 35 U.S.C. § 103(a)

While the Examiner did not base the rejection on 35 U.S.C. § 103, Applicants nonetheless submit that the prior art also does not render the pending claims obvious.

The Legal Standard for 35 U.S.C. § 103

The U.S. Patent and Trademark Office has the burden under 35 U.S.C. § 103 to establish a *prima facie* case of obviousness. *In re Warner et al.*, 379 F.2d 1011, 154 U.S.P.Q. 173, 177 (C.C.P.A. 1967), *In re Fine,* 837 F.2d 1071, 1074, 5 U.S.P.Q.2d 1596, 1598-99 (Fed. Cir. 1988). To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art and not based on applicants' disclosure. *In re Vaeck, 947 F.2d 488, 20 U.S.P.Q.2d 1438 (Fed. Cir. 1991).* The references cited by the Examiner do not meet all three criteria.

While the "teaching, suggestion, motivation test" should not be applied rigidly, it nonetheless provides insight into the obviousness inquiry. KSR Int'l v. Teleflex Inc., 82 U.S.P.Q.2d 1385, 1391, 1396 (2007). The prior art must provide one of ordinary skill in the art with the motivation to make the proposed modification needed to arrive at the claimed invention. *In re Geiger*, 815 F.2d 686, 2 U.S.P.Q.2d 1276 (Fed. Cir. 1987); *in re Lalu and Foulletier*, 747

PAGE 15/26 * RCVD AT 5/9/2008 5:26:04 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-6/46 * DNIS:2738300 * CSID:404 572 5135 * DURATION (mm-ss):06-12

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1255

F.2d 703, 705, 223 U.S.P.Q. 1257, 1258 (Fed. Cir. 1984). Claims for an invention are not *prima facie* obvious if the primary references do not suggest all elements of the claimed invention and the prior art does not suggest the modifications that would bring the primary references into conformity with the application claims. *In re Fritch*, 23 U.S.P.Q.2d, 1780 (Fed. Cir. 1992). *In re Laskowski*, 871 F.2d 115 (Fed. Cir. 1989). This is not possible when the claimed invention achieves more than what any or all of the prior art reference allegedly suggest, expressly or by reasonable implication.

It is clear that to establish a rejection under 35 U.S.C. § 103 the cited references must (1) recite each element of the claims, (2) provide one of skill in the art with the motivation to combine the cited reference as applications have done and (3) provide one of ordinary skill in the art with a reasonable expectation of success. The references cited by the Examiner clearly do not meet all three criteria and the current rejection of the claims-in-issue lack proper support.

Analysis

The *Magee* reference disclosed by the Examiner fails to disclose or render obvious under 35 U.S.C. § 103 the recitations that are present in independent Claims 1, 13 and 36.

Independent Claims 1 and 13

The *Magee* reference disclosed by the Examiner fails to disclose or suggest the claimed feature of monitoring the new program while allowing it to execute when the new program is different or not validated. In contrast, *Magee* discloses performing a matching or verification step prior to execution of an application. (See col. 6, lines 31-46; col. 11, lines 12-58; col. 12, lines 1-4, 52-56.) *Magee* actually teaches away from the methods recited in Claims 1 and 13 in that *Magee* states "if the hash value generated by the receiving processor does not match the hash value on the approved hash list, the system prevents the executable file data from executing . . . ." (*Magee*, col. 11, lines 54-57.) This step of preventing the executable file data from executing also is illustrated as the final step (step 522) in the process illustrated in Figure 5. In other words, there are no subsequent steps in the process allowing the unverified executable data to execute while being monitored as described in Claims 1 and 13.

Accordingly, Applicants submit that *Magee* fails to disclose or suggest the methods recited in Claims 1 and 13.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1256

Independent Claim 36

The *Magee* reference disclosed by the Examiner fails to disclose or suggest the claimed feature of "an execution module coupled to the detection module and operable for monitoring the program in response to the trigger intercepted by the detection module." Rather, *Magee* discloses a system where a single verification step is performed. For example, *Magee* describes granting program executability on a per-program basis based on a verification of data elements. (*Magee* col. 12, lines 1-4.) If the system in *Magee* is not able to verify the program attempting to execute, "the system prevents the executable file data from executing." Applicants submit that there is no disclosure or suggestion in *Magee* of an execution module that monitors a program in response to a trigger.

Accordingly, Applicants submit that *Magee* does not render Claim 36 obvious.

Dependent Claims 2-12, 14-20, and 37-40

Applicants respectfully submit that the above-identified dependent claims are allowable because the independent claims from which they depend are patentable over the cited documents. Applicants also respectfully submit that the recitations of these dependent claims are of patentable significance.

In view of the foregoing, Applicants respectfully request that the Examiner withdraw the pending rejections of dependent Claims 2-12, 14-20, and 37-40.

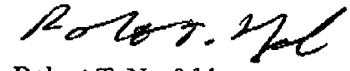
IPR2024-00027
CrowdStrike Exhibit 1002 Page 1257

RECEIVED
CENTRAL FAX CENTER
MAY 09 2008

Conclusion

In view of the arguments presented herein, Applicants respectfully request that the rejection in this matter be vacated, and that this application be returned to the Examiner with instructions to enter a notice of allowance.

Respectfully submitted,



Robert T. Neufeld
Reg. No. 48,394

KING & SPALDING LLP
1180 Peachtree Street
34th Floor
Atlanta, GA 30309
(404) 572-4600 (Telephone)
(404) 572-5134 (Facsimile)

PAGE 18/26 * RCVD AT 5/9/2008 5:26:04 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-6/46 * DNIS:2738300 * CSID:404 572 5135 * DURATION (mm-ss):06-12

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1258

RECEIVED
CENTRAL FAX CENTER
MAY 09 2008

APPENDIX 1

CLAIMS APPENDIX

1. (Original) A computer-implemented method for implementing security for a computing device, comprising the steps of:

identifying an allowed program that is permitted to execute on the computing device;

receiving a signal that a new program is going to be executed on the computing device;

suspending the execution of the new program on the computing device;

determining whether the new program is the same as the allowed program;

if the new program is the same as the allowed program, permitting the new program to execute on the computing device; and

if the new program is not the same as the allowed program, monitoring the new program while allowing it to execute on the computing device.

2. (Original) The method of Claim 1, wherein the step of receiving a signal that a new program is going to be executed is performed at the operating system kernel of the computing device.

3. (Original) The method of Claim 1, wherein the step of monitoring the new program while allowing it to execute is performed at the operating system kernel of the computing device.

4. (Original) The method of Claim1, wherein the step of determining whether the new program is the same as the allowed program comprises:

computing a checksum for the new program, and

comparing the new program's checksum to a checksum for the allowed program.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1259

5. (Original) The method of Claim 1, wherein the step of determining whether the new program is the same as the allowed program comprises determining whether the new program has been modified.

6. (Original) The method of Claim 1, wherein the step of monitoring the new program while allowing it to execute comprises controlling the files the program attempts to access during execution of the new program on the computing device.

7. (Original) The method of Claim 1, wherein the step of monitoring the new program while allowing it to execute comprises controlling the registry settings the new program attempts to access during execution of the new program on the computing device.

8. (Original) The method of Claim 1, wherein the step of monitoring the new program while allowing it to execute comprises controlling the network activity of the new program.

9. (Original) The method of Claim 1, wherein the step of monitoring the new program while allowing it to execute comprises determining whether a user has previously approved operation of the new program.

10. (Original) The method of Claim 1, further comprising the step of terminating the execution of the new program if the program performs suspicious activities.

11. (Original) The method of Claim 1, further comprising the step of taking remedial measures to protect the security of other computing device components if the new program is not the same as the allowed program.

12. (Original) A computer-readable medium having computer-executable instructions for performing the steps recited in Claim 1.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1260

13. (Original) A computer-implemented method for implementing security for a computing device comprising the steps of:

interrupting the loading of a new program for operation with the computing device;

validating the new program;

if the new program is validated, permitting the new program to continue loading and to execute in connection with the computing device;

if the new program is not validated, monitoring the new program while it loads and executes in connection with the computing device.

14. (Original) The method of Claim 13, wherein the step of interrupting the loading of a new program comprises:

intercepting a signal from the computing device's operating system that the new program is loading, and

suspending the loading of the new program.

15. (Original) The method of Claim 13, wherein the step of monitoring the new program comprises intercepting a signal from the computing device's operating system.

16. (Original) The method of Claim 13, wherein the step of validating the new program comprises determining whether the new program corresponds with an approved program.

17. (Original) The method of Claim 13, wherein the step of validating the new program comprises comparing a checksum for the new program with a previously determined checksum.

18. (Original) The method of Claim 13, wherein the step of validating the new program comprises analyzing characteristics of the new program.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1261

19. (Original) The method of Claim 13, wherein the step of monitoring the new program comprises controlling the files the new program attempts to access during execution of the new program.

20. (Original) A computer-readable medium having computer-executable instructions for performing the steps recited in Claim 13.

[This section is intentionally left blank.]

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1262

The rejection of Claims 21-35 is not appealed.

[This section is intentionally left blank.]

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1263

36. (Original) A system for managing security of a computing device comprising:

a pre-execution module operable for receiving notice from the computing device's operating system that a program is being loaded onto the computing device;

a validation module coupled to the pre-execution monitor operable for determining whether the program is valid;

a detection module coupled to the pre-execution monitor operable for intercepting a trigger from the computing device's operating system; and

an execution module coupled to the detection module and operable for monitoring the program in response to the trigger intercepted by the detection module.

37. (Original) The system of Claim 36, wherein the pre-execution module is further operable for suspending loading of the program onto the computing device.

38. (Original) The system of Claim 36, wherein the pre-execution module is further operable for retrieving validation data.

39. (Original) The system of Claim 36, wherein the execution module is further operable for deciding whether to terminate the trigger intercepted by the detection module.

40. (Original) The system of Claim 36, wherein the execution module is further operable for retrieving authorization data and deciding how to respond to the trigger intercepted by the detection module.

[This section is intentionally left blank.]

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1264

APPENDIX 2

EVIDENCE APPENDIX

None.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1265

APPENDIX 3

RELATED PROCEEDINGS APPENDIX

None.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1266

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.
10/336,299	01/03/2003	Thomas James Satterlee	05456.105038	7850

69151 7590 07/24/2008

KING & SPALDING, LLP
INTELLECTUAL PROPERTY DEPT. - PATENTS
1180 PEACHTREE STREET, N.E.
ATLANTA, GA 30309-3521

EXAMINER	
JACKSON, JENISE E	
ART UNIT	PAPER NUMBER
2139	

MAIL DATE	DELIVERY MODE
07/24/2008	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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1267

Office Action Summary	Application No.	Applicant(s)	
	10/336,299	SATTERLEE ET AL.	
	Examiner	Art Unit	
	JENISE E. JACKSON	2139	

-- 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 May 2008*.

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) *1-40* 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) *1-40* 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: _____.

DETAILED ACTION

1. In view of the Applicant's arguments filed on 5/9/08, PROSECUTION IS HEREBY REOPENED. A new art rejection has been applied as set forth below.

To avoid abandonment of the application, appellant must exercise one of the following two options:

(1) file a reply under 37 CFR 1.111 (if this Office action is non-final) or a reply under 37 CFR 1.113 (if this Office action is final); or,

(2) initiate a new appeal by filing a notice of appeal under 37 CFR 41.31 followed by an appeal brief under 37 CFR 41.37. The previously paid notice of appeal fee and appeal brief fee can be applied to the new appeal. If, however, the appeal fees set forth in 37 CFR 41.20 have been increased since they were previously paid, then appellant must pay the difference between the increased fees and the amount previously paid.

A Supervisory Patent Examiner (SPE) has approved of reopening prosecution by signing below:

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

3. Claims 36-40 are rejected under 35 U.S.C. 102(e) as being anticipated by McGee(6,694,434).

4. As per claim 36, McGee discloses a system for managing security of a computing device a pre-execution module operable for receiving notice from the computing device's operating system that a program is being loaded onto the computing device; a validation module coupled to the pre-execution monitor operable for determining whether the program is valid(see col. 6, lines 3-19); a detection module coupled to the pre-execution monitor operable for intercepting a trigger from the computing device's operating system(see col. 11, lines 3-12, 38-67, col. 12, lines 1-51); and an execution module coupled to the detection module and operable for monitoring the program in response to the trigger intercepted by the detection module(see col. 12, lines 15-18).

5. As per claim 37, McGee discloses wherein the pre-execution module is further operable for suspending loading of the program onto the computing device(see col. 8, lines 11-40).

6. As per claim 38, McGee discloses wherein the pre-execution module is further operable for retrieving validation data(see col. 6, lines 3-19).

7. As per claim 39, McGee discloses wherein the execution module is further operable for deciding whether to terminate the trigger intercepted by the detection module(see col. 4, lines 40-61).

8. As per claim 40, McGee discloses wherein the execution module is further operable for retrieving authorization data and deciding how to respond to the trigger intercepted by the detection module(see col. 6, lines 3-19).

Claim Rejections - 35 USC § 103

9. 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.

10. Claims 1-35 are rejected under 35 U.S.C. 103(a) as being unpatentable over McGee et al(6,694,434) in view of Bush et al(2002/0184520) .

11. As per claim 1, McGee et al. discloses identifying an allowed program that is permitted to execute on the computing device(see col. 4, lines 1-23); receiving a signal that a new program is going to be executed on the computing device(see col. 6, lines 39-42, 46-50); suspending the execution of the new program on the computing device; determining whether the new program is the same as the allowed program(col. 8, lines 11-40); if the new program is the same as the allowed program, permitting the new program to execute on the computing device(see col. 12, lines 52-63). McGee does not disclose and if the new program is not the same as the allowed program, monitoring the new program while allowing it to execute on the computing device. Bush discloses if the new program is not the same as the allowed program, monitoring the new program while allowing it to execute on the computing device[0007-0009]. It would have been obvious to one of ordinary skill in the art to include monitoring the new program while allowing it to execute on the computing device of Bush with McGee, because often there is a need for flexibility when dealing with untrusted code(i.e. new)[0009 of Bush]. Bush discloses there may be a need to permit untrusted code to have some defined access to the trusted computer

resources, such as to particular data managed by the trusted code or to particular methods of the trusted code[0009 of Bush].

12. As per claim 2, McGee discloses wherein the step of receiving a signal that a new program is going to be executed is performed at the operating system kernel of the computing device(see col. 11, lines 7-12, col. 12, lines 52-63).

13. As per claim 3, McGee discloses wherein the step of monitoring the new program while allowing it to execute is performed at the operating system kernel of the computing device(see col. 6, lines 19-50, col. 11, lines 7-12).

14. As per claim 4, McGee discloses wherein the step of determining whether the new program is the same as the allowed program(see col. 12, lines 52-63), computing a checksum for the new program, and comparing the new program's checksum to a checksum for the allowed program(col. 4, lines 35-49).

15. As per claim 5, McGee discloses wherein the step of determining whether the new program is the same as the allowed program comprises determining whether the new program has been modified (col. 9, lines 12-37).

16. As per claim 6, McGee discloses wherein the step of monitoring the new program while allowing it to execute includes controlling the files the program attempts to access during execution of the new program on the computing device(see col. 11, lines 3-36, col. 12, lines 19-63).

17. As per claim 7, McGee discloses wherein the step of monitoring the new program while allowing it to execute includes controlling the registry settings the new program attempts to access during execution of the new program on the computing device(col. 6, lines 51-67, col. 12,

lines 52-63).

18. As per claim 8, McGee discloses wherein the step of monitoring the new program while allowing it to execute includes controlling the network activity of the new program(col. 6, lines 51-67, col. 8, lines 11-40).

19. As per claim 9, McGee discloses wherein the step of monitoring the new program while allowing it to execute includes determining whether a user has previously approved operation of the new program(see col. 12, lines 19-63).

20. As per claim 10, McGee discloses the step of terminating the execution of the new program if the program performs suspicious activities(see col. 4, lines 40-61).

21. As per claim 11, McGee discloses the step of taking remedial measures to protect the security of other computing device components if the new program is not the same as the allowed program(col. 8, lines 11-40).

22. As per claim 12, McGee discloses a computer-readable medium having computer-executable instructions for performing the steps recited in claim 1(see col. 10, lines 23-38).

23. As per claim 13, McGee discloses a computer-implemented method for implementing security for a computing device(see col. 10, lines 23-38), interrupting the loading of a new program for operation with the computing device; validating the new program; if the new program is validated, permitting the new program to continue loading and to execute in connection with the computing device. McGee does not disclose if the new program is not validated, monitoring the new program while it loads and executes in connection with the computing device. Bush discloses if the new program is not the same as the allowed program, monitoring the new program while allowing it to execute on the computing device[0007-0009].

It would have been obvious to one of ordinary skill in the art to include monitoring the new program while allowing it to execute on the computing device of Bush with McGee, because often there is a need for flexibility when dealing with untrusted code(i.e. new)[0009 of Bush]. Bush discloses there may be a need to permit untrusted code to have some defined access to the trusted computer resources, such as to particular data managed by the trusted code or to particular methods of the trusted code[0009 of Bush].

24. As per claim 14, McGee discloses wherein the step of interrupting the loading of a new program comprises: intercepting a signal from the computing device's operating system that the new program is loading, and suspending the loading of the new program(col. 8, lines 11-40).

25. As per claim 15, McGee discloses wherein the step of monitoring the new program includes intercepting a signal from the computing device's operating system((see col. 11, lines 7-12, col. 12, lines 52-63).

26. As per claim 16, McGee discloses wherein the step of validating the new program includes determining whether the new program corresponds with an approved program(col. 6, lines 19-50).

27. As per claim 17, McGee discloses wherein the step of validating the new program includes comparing a checksum for the new program with a previously determined checksum((col. 4, lines 35-49).

28. As per claim 18, McGee discloses wherein the step of validating the new program comprises analyzing characteristics of the new program(col. 8, lines 11-40).

29. As per claim 19, McGee discloses wherein the step of monitoring the new program includes controlling the files the new program attempts to access during execution of the new

program(see col. 11, lines 3-36, col. 12, lines 19-63).

30. As per claim 20, McGee discloses a computer-readable medium having computer-executable instructions for performing the steps recited in claim 13(see col. 10, lines 23-38).

31. As per claim 21, McGee discloses a computer-implemented method for performing security for a computing device during a pre-execution phase(col. 10, lines 23-38) identifying an allowed program that is permitted to execute with the computing device(see col. 4, lines 1-23); receiving a signal that a new program is being loaded for execution with the computing device(see col. 11, lines 7-12, col. 12, lines 52-63); suspending the loading of the new program; comparing the new program to the allowed program(see col. 8, lines 11-40); and determining whether the new program is valid(col. 12, lines 52-63).

32. As per claim 22, McGee discloses wherein the step of receiving a signal includes receiving a signal from the computing device's operating system kernel(col. 11, lines 7-12, col. 12, lines 52-63).

33. As per claim 23, McGee discloses wherein the step of comparing the new program to the allowed program includes comparing a checksum of the new program to a checksum of the allowed program(col. 4, lines 35-49).

34. As per claim 24, McGee discloses wherein the step of determining whether the new program is valid includes determining whether a checksum of the new program corresponds to a checksum of the allowed program(see col. 4, lines 35-49).

35. As per claim 25, McGee discloses a computer-readable medium having computer-executable instructions for performing the steps recited in claim 21(col. 10, lines 23-38).

36. As per claim 26, McGee discloses a computer-implemented method for performing

security for a computing device during an execution phase(see col. 10, lines 23-38), intercepting a call to open a file from the computing device's operating system(see col. 11, lines 7-12); determining whether the file is restricted; if the file is restricted, taking remedial action to protect the computing device; and if the file is not restricted, permitting the call to open the file(see col. 8, lines 11-40).

37. As per claim 27, McGee discloses the step of receiving an instruction to monitor a call from an executing program(see col. 11, lines 7-12).

38. As per claim 28, McGee discloses the step of defining files that are restricted from access(see col. 11, lines 17-35).

39. As per claim 29, McGee discloses, wherein the step of defining files that are restricted includes designating a file that cannot be accessed in response to the call(see col. 11, lines 3-36).

40. As per claim 30, McGee discloses wherein the step of defining files that are restricted includes designating a file that cannot be modified in response to the call(see col. 11, lines 3-36, col. 13, lines 8-20).

41. As per claim 31, McGee discloses the step of alerting a security management module of the presence of a malicious program(see col. 13, lines 8-20).

42. As per claim 32, McGee discloses wherein the step of taking remedial action includes terminating the call(see col. 8, lines 11-40).

43. As per claim 33, McGee discloses wherein the step of taking remedial action includes alerting another security module of the existence of the call(see col. 8, lines 11-40, col. 13, lines 8-20).

44. As per claim 34, McGee discloses wherein the step of taking remedial action includes

notifying the user of the existence of the call(see col. 8, lines 11-40, col. 11, lines 7-12).

45. As per claim 35, McGee discloses a computer-readable medium having computer-executable instructions for performing the steps recited in claim 26(see col. 10, lines 23-38).

Response to Applicant

46. The Applicant states that McGee does not disclose monitoring of the new program while allowing it to execute on the computing device if the new program is not validated. The Applicants arguments were persuasive and thus new art has been applied to the claims in regards to this limitation.

47. The Applicant states that McGee does not disclose an execution module that monitors a program in response to a trigger. The Examiner disagrees with the Applicant. McGee discloses an output signal is generated to trigger a request for user approval to indicate approval of the calling application(see col. 6, lines 40-42). McGee discloses if the names nor hash values match, the output signal is generated to trigger a request for user approval(see col. 6, lines 42-44). McGee discloses that the verifier also receives approval data entered by a user to indicate that a particular application has been approved for execution of privileges(see col. 6, lines 46-50).

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to JENISE E. JACKSON whose telephone number is (571)272-3791. The examiner can normally be reached on Increased Flex time, but generally in the office M-Fri(8-4:30)..

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kristine Kincaid can be reached on (571) 272-4063. 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.

July 18, 2008
/J. E. J./

Examiner, Art Unit 2139

/Kristine Kincaid/

Supervisory Patent Examiner, Art Unit 2139

Notice of References Cited	Application/Control No. 10/336,299	Applicant(s)/Patent Under Reexamination SATTERLEE ET AL.	
	Examiner JENISE E. JACKSON	Art Unit 2139	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-2002/0184520	12-2002	Bush et al.	713/200
	B	US-2003/0177394	09-2003	Dozortsev, Dmitri	713/201
	C	US-			
	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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1279

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10336299	SATTERLEE ET AL.
	Examiner	Art Unit
	JENISE E JACKSON	2139

✓	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	07/18/2008								
	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	✓								
	32	✓								
	33	✓								
	34	✓								
	35	✓								
	36	✓								

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1280

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10336299	SATTERLEE ET AL.
	Examiner	Art Unit
	JENISE E JACKSON	2139

✓	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	07/18/2008								
	37	✓								
	38	✓								
	39	✓								
	40	✓								

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1281

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10336299	SATTERLEE ET AL.
	Examiner	Art Unit
	JENISE E JACKSON	2139

SEARCHED			
Class	Subclass	Date	Examiner
713	164-165, 167	7/18/08	JJ
726	22, 24, 26	7/18/08	JJ

SEARCH NOTES		
Search Notes	Date	Examiner
Updated West text search	7/18/08	JJ
Updated West class and text search	7/18/08	JJ
Updated Inventor and Assignee search	7/18/08	JJ
Consulted Matt Heneghan	7/15/08	JJ

INTERFERENCE SEARCH			
Class	Subclass	Date	Examiner

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1282

WEST Search History for Application 10336299

Creation Date: 2008071814:37

Query	DB	Op.	Plur.	Thes.	Date
(allowed or approved)with(program or code or software or executable or application)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
(determin$4)near5(new)with(program or software or code or executable or application)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
(pre-approv$ or approv$4 or allow$4)with(list or database) and ((determin$4)near5(new)with(program or software or code or executable or application))	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
((pre-approv$ or approv$4 or allow$4)with(list or database) and (determin$4)near5(new)with(program or software or code or executable or application)) and sandbox	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
Monitor$4 near5(kernel)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
(validat$4 or authenticat$4 or authoriz$4 or verif$4)near5(new)near4(program or application or software or code or executable)with(approv$4 or allow$4)near5(program or application or code or software or executable)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
(compar$4)near5(new)with(program or application or software or code or executable)near5(list or database)with(approv$4 or allow$4 or pre-approv$4)	PGPB, USPT, EPAB, JPAB,	OR	YES		07-18-2008

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1283

	DWPI, TDBD				
(monitor$4)near5(new)with(program or software or application or code or executable) and (allow$4 or approv$4)near5(computer)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
((monitor$4)near5(new)with(program or software or application or code or executable) and (allow$4 or approv$4)near5(computer)) and sandbox	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
(execut$4)near5(new)with(program or software or executable or code or application)near5(sandbox)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
TPM and sandbox	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
(compar$4)with(new adj program)with(approv$4 or allow$4)near4(program)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
(list)with(approv$4 or allow$4)near5(program)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
((list)with(approv$4 or allow$4)near5(program)) and sandbox	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
		OR	YES		07-18-2008

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1284

(new)near5(program or application or software or code or executable)same(allow$4 or approv$4)near5(program or application or software or code or executable)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD				
((new)near5(program or application or software or code or executable)same(allow$4 or approv$4)near5(program or application or software or code or executable)) and sandbox	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
726/1.ccls.and (pre-approv? or approv? or allow?)near5(program or executable or software)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
726/2.ccls.and (pre-approv? or approv? or allow?)near5(program or executable or software)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
726/14.ccls.and (pre-approv? or approv? or allow?)near5(program or executable or software)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
726/22.ccls.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
726/24.ccls.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
726/26.ccls.	PGPB, USPT, EPAB, JPAB,	OR	YES		07-18-2008

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1285

	DWPI, TDBD				
713/161.ccls.and (pre-approv? or approv? or allow?)near5(program or executable or software)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
713/164.ccls.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
713/165.ccls.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
713/182.ccls.and (pre-approv? or approv? or allow?)near5(program or executable or software)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
713/187.ccls.and (pre-approv? or approv? or allow?)near5(program or executable or software)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
713/188.ccls.and (pre-approv? or approv? or allow?)near5(program or executable or software)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
713/167.ccls.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
Thomas near James near Satterlee.in.		OR	YES		07-18-2008

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1286

	PGPB, USPT, EPAB, JPAB, DWPI, TDBD				
William near Frank near Hackenberger.in.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
Internet near Security near Systems.as. and (execut?)near4(program)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
TPM and sandbox	PGPB	OR			07-18-2008
sandbox	PGPB	OR			07-18-2008
(allow$4)near5(new)with(program or application or executable or software or code)with(run$4 or execut$4) and (sandbox)	PGPB	OR			07-18-2008
(sandbox) and (monitor$4)with(new)with(program or software or code or executable or application)	PGPB	OR			07-18-2008
(sandbox) and Gong.in.	PGPB	OR			07-18-2008
(sandbox) and (allow$4)with(execut$4)near5(computer)	PGPB	OR			07-18-2008
(sandbox) and (applet)	PGPB	OR			07-18-2008
(allowed or approv$4 or authoriz$4)near5(program or software or application or executable or code)near5(run$4 or execut$4)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR			07-18-2008
(new)near5(program or executable or software or application or code)near5(compar$4 or validat$4 or authenticat$4) and ((allowed or approv$4 or authoriz$4)near5(program or software or application or executable or code)near5(run$4 or execut$4))	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR			07-18-2008
((new)near5(program or executable or software or application or code)near5(compar$4 or validat$4 or	PGPB, USPT,	OR			07-18-2008

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1287

authenticat$4) and (allowed or approv$4 or authoriz$4)near5(program or software or application or executable or code)near5(run$4 or execut$4)) and sandbox	EPAB, JPAB, DWPI, TDBD				
(new)near5(program or executable or software or application or code)near5(compar$4 or validat$4 or authenticat$4)with(allow$4 or approv$4)near5(program or software or application or executable or code)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR			07-18-2008
(new or unknown)with(program or software or application or code or applet or activex)near5(non-allow$4 or non-approv$4)with(program or software or application or code or applet or activex)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR			07-18-2008
((allowed or approv$4 or authoriz$4)near5(program or software or application or executable or code)near5(run$4 or execut$4)) and (sandbox)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR			07-18-2008
(new or unknown)near5(no near4 validat$4 or no near4 authoriz$4 or no near5 authenticat$4)with(program or software or application or code) and ((allow$4)near5(new)with(program or application or executable or software or code)with(run$4 or execut$4) and sandbox)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR			07-18-2008
(new or unknown)near5(no near4 validat$4 or no near4 authoriz$4 or no near5 authenticat$4)with(program or software or application or code) and ((allowed or approv$4 or authoriz$4)near5(program or software or application or executable or code)near5(run$4 or execut$4) and sandbox)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR			07-18-2008

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1288

PATENTS

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	)	
	)	Attorney Docket No. **05456.105038**
Thomas James Satterlee et al.	)	
	)	Group Art Unit: **2139**
Application No. **10/336,299**	)	
	)	Confirmation No.: **7850**
Filed: **January 3, 2003**	)	
	)	Examiner: **Jackson, Jenise E.**
For: **System and Method for the Managed Security Control of Processes on a Computer**	)	

APPEAL BRIEF

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

Sir:

Further to the Notice of Appeal filed concurrently herewith in the above referenced application, Appellants hereby submit this brief under 37 C.F.R. § 1.191 to appeal the Examiner's rejection of this application, as reported in the Office Action mailed on July 24, 2008.

Please apply the $500.00 appeal brief fee paid with the previous appeal brief submitted September 4, 2007 and the $10 additional fee paid with the previous appeal brief submitted May 9, 2008.

The Commissioner is authorized to charge any additional fee required for this Appeal Brief, or to credit any overpayment, to Deposit Account No. 09-0457.

Respectfully submitted,

/ Robert T. Neufeld/

Robert T. Neufeld
Reg. No. 48,394

I hereby certify that this correspondence is being electronically transmitted via EFS-web to the Commissioner for Patents on October 24, 2008.

/Robert T. Neufeld/
Robert T. Neufeld, Reg. No. 48,394

Table of Contents

Real Party in Interest 3

Related Appeals and Interferences 4

Status of Claims 5

Status of Amendments 6

Summary of Claimed Subject Matter 7

Grounds of Rejection to be Reviewed on Appeal 8

Argument 9

Conclusion 12

APPENDIX 1 13

APPENDIX 2 14

APPENDIX 3 15

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1290

Real Party in Interest

The real party in interest is IBM Internet Security Services, formerly operating as Internet Security Systems, Inc. IBM Internet Security Services is a wholly owned subsidiary of International Business Machines Corporation.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1291

Related Appeals and Interferences

None.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1292

Status of Claims

Claims 1-40 stand rejected in the most recent Office Action of July 24, 2008.

The rejection of Claim 3 is appealed.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1293

Status of Amendments

No amendments have been filed subsequent to the final rejection mailed on July 24, 2008.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1294

Summary of Claimed Subject Matter

The method of dependent Claim 3 includes the following steps from Claims 1 and 3: (1) identifying an allowed program that is permitted to execute on the computing device (page 11, lines 30 to 31; page 12, lines 18 to 22); (2) receiving a signal that a new program is going to be executed on the computing device (page 5, lines 14 to 16; page 13, lines 26 to 27); (3) suspending the execution of the new program on the computing device (page 14, lines 7 to 8; page 6, lines 1 to 3); (4) determining whether the new program is the same as the allowed program (page 5, lines 16 to 17; page 14, lines 12 to 15); (5) if the new program is the same as the allowed program, permitting the new program to execute on the computing device (page 5, lines 17 to 19; page 14, lines 15 to 17); (6) if the new program is not the same as the allowed program, monitoring the new program while allowing it to execute on the computing device (page 5, lines 21 to 23; page 14 lines 17 to 20); and (7) wherein the step of monitoring the new program while allowing it to execute is performed at the operating system kernel of the computing device (page 9, lines 7-16; page 10, lines 14-17; page 11, lines 1-3).

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1295

Grounds of Rejection to be Reviewed on Appeal

The following issue is presented on appeal:

(1) Whether, under 35 U.S.C. § 103, Claim 3 is obvious in view of the combination of U.S. Patent No. 6,694,434 to McGee ("McGee") and U.S. Published Patent Application No. 2002/0184520 to Bush ("Bush").

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1296

Argument

Rejection of claims under 35 U.S.C. § 103(a)

Applicants submit that the prior art of record does not render pending Claim 3 obvious.

The Legal Standard for 35 U.S.C. § 103

The U.S. Patent and Trademark Office has the burden under 35 U.S.C. § 103 to establish a *prima facie* case of obviousness. *In re Warner et al.*, 379 F.2d 1011, 154 U.S.P.Q. 173, 177 (C.C.P.A. 1967), *In re Fine,* 837 F.2d 1071, 1074, 5 U.S.P.Q.2d 1596, 1598-99 (Fed. Cir. 1988). To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art and not based on applicants' disclosure. *In re Vaeck, 947 F.2d 488, 20 U.S.P.Q.2d 1438 (Fed. Cir. 1991).* The references cited by the Examiner do not meet all three criteria.

While the "teaching, suggestion, motivation test" should not be applied rigidly, it nonetheless provides insight into the obviousness inquiry. *KSR Int'l v. Teleflex Inc.*, 82 U.S.P.Q.2d 1385, 1391, 1396 (2007). The prior art must provide one of ordinary skill in the art with the motivation to make the proposed modification needed to arrive at the claimed invention. *In re Geiger*, 815 F.2d 686, 2 U.S.P.Q.2d 1276 (Fed. Cir. 1987); *in re Lalu and Foulletier*, 747 F.2d 703, 705, 223 U.S.P.Q. 1257, 1258 (Fed. Cir. 1984). Claims for an invention are not *prima facie* obvious if the primary references do not suggest all elements of the claimed invention and the prior art does not suggest the modifications that would bring the primary references into conformity with the application claims. *In re Fritch*, 23 U.S.P.Q.2d, 1780 (Fed. Cir. 1992). *In re Laskowski*, 871 F.2d 115 (Fed. Cir. 1989). This is not possible when the claimed invention achieves more than what any or all of the prior art reference allegedly suggest, expressly or by reasonable implication.

It is clear that to establish a rejection under 35 U.S.C. § 103 the cited references must (1) recite each element of the claims, (2) provide one of skill in the art with the motivation to combine the cited reference as applications have done and (3) provide one of ordinary skill in the

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1297

art with a reasonable expectation of success. The references cited by the Examiner clearly do not meet all three criteria and the current rejection of the claims-in-issue lack proper support.

Analysis

The McGee and Bush references fail to disclose or render obvious under 35 U.S.C. § 103 the recitations that are present in Claim 3, which depends from Claim 1.

Dependent Claim 3

In the Office Action, the Examiner rejected independent Claim 1 as obvious in view of the combination of the McGee and Bush references. The Examiner further rejected dependent Claim 3 based on additional teachings in McGee.

In the Office Action, the Examiner acknowledges that McGee does not disclose the last step of Claim 1 where a program is not pre-approved and is therefore monitored while it is allowed to execute. The Examiner asserts that Bush discloses the last step of Claim 1 in that paragraphs 0007 - 0009 of Bush describe a "sandbox method" where trusted code is permitted to execute with full access to a computer system's resources and untrusted code is permitted to execute, but only with limited access to a computer system's resources. The Bush reference goes on to describe using a virtual machine, which is separate from the operating system, to process untrusted code. (See paragraphs 0010 and 0012.)

Claim 3, which depends from Claim 1, provides further detail concerning the last step of Claim 1 in that the monitoring of the new program while allowing it to execute "is performed at the operating system kernel of the computing device." The Examiner contends that column 6, lines 19-50 and column 11, lines 7-12 of McGee disclose the step recited in dependent Claim 3.

Applicants respectfully submit that McGee does not disclose monitoring an unapproved program from the operating system kernel while allowing the program to execute. The cited portions of the McGee reference concern performing a matching or verification step prior to execution of an application. (See col. 6, lines 31-46; col. 11, lines 12-58; col. 12, lines 1-4, 52-56.) McGee actually teaches away from the method recited in Claim 3 in that McGee states "if the hash value generated by the receiving processor does not match the hash value on the approved hash list, the system prevents the executable file data from executing . . . ." (McGee, col. 11, lines 54-57.) This step of preventing the executable file data from executing also is

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1298

illustrated as the final step (step 522) in the process illustrated in Figure 5. In other words, there are no subsequent steps in the process allowing the unverified executable data to execute while being monitored at the operating system kernel as described in Claim 3.

Furthermore, Applicants submit that the disclosures in Bush of a "sandbox method" and using a virtual machine do not teach the method recited in Claim 3. In particular, Claim 3 recites "monitoring the new program at an operating system kernel of the computing device". Allowing the program to execute and monitoring the new program at the operating system kernel are distinct from the sandbox or virtual machine methods disclosed in Bush where the untrusted code is processed in an artificial environment separate from the trusted code. As explained in the Background portion of the specification of the present application, "[v]irtual execution is limited in its ability to detect harmful code because it does not actually execute every process of the code." (See page 2, line 30 - page 3, line 1.)

In contrast to the virtual execution of untrusted code described in Bush, Claim 3 recites allowing the unknown program to execute, but monitoring the execution at the operating system kernel. Therefore, Applicants submit that Claim 3 is patentably distinct from the art of record.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1299

Conclusion

In view of the arguments presented herein, Applicants respectfully request that the rejection in this matter be vacated, and that this application be returned to the Examiner with instructions to enter a notice of allowance.

Respectfully submitted,

/Robert T. Neufeld/

Robert T. Neufeld
Reg. No. 48,394

KING & SPALDING LLP
1180 Peachtree Street
34th Floor
Atlanta, GA 30309
(404) 572-4600 (Telephone)
(404) 572-5134 (Facsimile)

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1300

APPENDIX 1

CLAIMS APPENDIX

1. (Original) A computer-implemented method for implementing security for a computing device, comprising the steps of:

identifying an allowed program that is permitted to execute on the computing device;

receiving a signal that a new program is going to be executed on the computing device;

suspending the execution of the new program on the computing device;

determining whether the new program is the same as the allowed program;

if the new program is the same as the allowed program, permitting the new program to execute on the computing device; and

if the new program is not the same as the allowed program, monitoring the new program while allowing it to execute on the computing device.

3. (Original) The method of Claim 1, wherein the step of monitoring the new program while allowing it to execute is performed at the operating system kernel of the computing device.

The rejection of Claims 1, 2 and 4-40 are not appealed at this time.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1301

APPENDIX 2

EVIDENCE APPENDIX

None.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1302

APPENDIX 3

RELATED PROCEEDINGS APPENDIX

None.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1303

Electronic Acknowledgement Receipt

EFS ID:	4176483
Application Number:	10336299
International Application Number:	
Confirmation Number:	7850
Title of Invention:	System and method for the managed security control of processes on a computer system
First Named Inventor/Applicant Name:	Thomas James Satterlee
Customer Number:	69151
Filer:	Robert Timothy Neufeld
Filer Authorized By:	
Attorney Docket Number:	05456.105038
Receipt Date:	24-OCT-2008
Filing Date:	03-JAN-2003
Time Stamp:	19:21:57
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Notice of Appeal Filed	END9_20075119_US2_NotApp_10_24_08.pdf	75771 84c36ff206e5fb5f6fe1ad4588dc224af7f4b1fa	no	1

Warnings:	
Information:	

2	Appeal Brief Filed	END9_20075119_US2_AppBrf_10_24_08.pdf	126741 9aea0e9fdfd5ee87ccbdfefb2b75ac31ef4f3e97	no	15

Warnings:	
Information:	
Total Files Size (in bytes):	202512

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.

PATENTS

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	**Thomas James Satterlee et al.**		Atty Docket: **05456.105038**
Application No.:	**10/336,299**		Art Unit: **2139**
Filing Date:	**January 3, 2003**		Examiner: **Jackson, Jenise E.**
Title:	**System and Method for the Managed Security Control of Processes of a Computer**		Confirmation No.: **7850**

NOTICE OF APPEAL

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

Sir:

Applicants hereby appeal to the Board of Patent Appeals and Interferences from the last decision of the Examiner mailed July 24, 2008.

Please apply the $500 Notice of Appeal fee previously paid with the notice submitted on July 2, 2007 and the $10 additional fee previously paid with the notice submitted on March 10, 2008.

The Commissioner is authorized to charge any additional fee required for this Notice of Appeal, or to credit any overpayment, to Deposit Account No. 09-0475. A duplicate of this paper is enclosed.

Respectfully submitted,

/ Robert T. Neufeld/

KING & SPALDING LLP
1180 Peachtree Street, N.E., 34th Floor
Atlanta, Georgia 30309-3521
(404) 572-4600

Robert T. Neufeld
Attorney for Applicants
Reg. No. 48,394

I hereby certify that this correspondence is being electronically transmitted via EFS-Web to the Commissioner for Patents on October 24, 2008.

/Robert T. Neufeld/
Robert T. Neufeld, Reg. No. 48,394

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.
10/336,299	01/03/2003	Thomas James Satterlee	05456.105038	7850

69151 7590 01/16/2009

KING & SPALDING, LLP
INTELLECTUAL PROPERTY DEPT. - PATENTS
1180 PEACHTREE STREET, N.E.
ATLANTA, GA 30309-3521

EXAMINER	
JACKSON, JENISE E	
ART UNIT	PAPER NUMBER
2439	

MAIL DATE	DELIVERY MODE
01/16/2009	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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1307

Office Action Summary	Application No.	Applicant(s)	
	10/336,299	SATTERLEE ET AL.	
	Examiner	Art Unit	
	JENISE E. JACKSON	2439	

-- 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 *24 October 2008*.

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) *1-40* 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) *1-40* 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: _____.

DETAILED ACTION

1. In view of the Applicant's arguments filed on 10/24/08, PROSECUTION IS HEREBY REOPENED. A new art rejection has been applied as set forth below.

To avoid abandonment of the application, appellant must exercise one of the following two options:

(1) file a reply under 37 CFR 1.111 (if this Office action is non-final) or a reply under 37 CFR 1.113 (if this Office action is final); or,

(2) initiate a new appeal by filing a notice of appeal under 37 CFR 41.31 followed by an appeal brief under 37 CFR 41.37. The previously paid notice of appeal fee and appeal brief fee can be applied to the new appeal. If, however, the appeal fees set forth in 37 CFR 41.20 have been increased since they were previously paid, then appellant must pay the difference between the increased fees and the amount previously paid.

A Supervisory Patent Examiner (SPE) has approved of reopening prosecution by signing below:

/Kambiz Zand/

Supervisory Patent Examiner, Art Unit 2434

Response to Applicant

46. The Applicant states that McGee nor Bush discloses monitoring of the new program while allowing it to execute is performed at the operating system kernel of the computing device as it is claimed in claims 3 and 22. The Applicants arguments were persuasive and thus new art has been applied to the claims in regards to this limitation.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

3. Claims 36-40 are rejected under 35 U.S.C. 102(e) as being anticipated by McGee(6,694,434).

4. As per claim 36, McGee discloses a system for managing security of a computing device a pre-execution module operable for receiving notice from the computing device's operating system that a program is being loaded onto the computing device; a validation module coupled to the pre-execution monitor operable for determining whether the program is valid(see col. 6, lines 3-19); a detection module coupled to the pre-execution monitor operable for intercepting a trigger from the computing device's operating system(see col. 11, lines 3-12, 38-67, col. 12, lines 1-51); and an execution module coupled to the detection module and operable for monitoring the program in response to the trigger intercepted by the detection module(see col. 12, lines 15-18).

5. As per claim 37, McGee discloses wherein the pre-execution module is further operable for suspending loading of the program onto the computing device(see col. 8, lines 11-40).

6. As per claim 38, McGee discloses wherein the pre-execution module is further operable for retrieving validation data(see col. 6, lines 3-19).

7. As per claim 39, McGee discloses wherein the execution module is further operable for deciding whether to terminate the trigger intercepted by the detection module(see col. 4, lines 40-61).

8. As per claim 40, McGee discloses wherein the execution module is further operable for retrieving authorization data and deciding how to respond to the trigger intercepted by the detection module(see col. 6, lines 3-19).

Claim Rejections - 35 USC § 103

9. 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.

10. Claims 1-2, 4-21, 23-35 are rejected under 35 U.S.C. 103(a) as being unpatentable over McGee et al(6,694,434) in view of Bush et al(2002/0184520) .

11. As per claim 1, McGee et al. discloses identifying an allowed program that is permitted to execute on the computing device(see col. 4, lines 1-23); receiving a signal that a new program is going to be executed on the computing device(see col. 6, lines 39-42, 46-50); suspending the execution of the new program on the computing device; determining whether the new program is the same as the allowed program(col. 8, lines 11-40); if the new program is the same as the allowed program, permitting the new program to execute on the computing device(see col. 12, lines 52-63). McGee does not disclose and if the new program is not the same as the allowed

program, monitoring the new program while allowing it to execute on the computing device. Bush discloses if the new program is not the same as the allowed program, monitoring the new program while allowing it to execute on the computing device[0007-0009]. It would have been obvious to one of ordinary skill in the art to include monitoring the new program while allowing it to execute on the computing device of Bush with McGee, because often there is a need for flexibility when dealing with untrusted code(i.e. new)[0009 of Bush]. Bush discloses there may be a need to permit untrusted code to have some defined access to the trusted computer resources, such as to particular data managed by the trusted code or to particular methods of the trusted code[0009 of Bush].

12. As per claim 2, McGee discloses wherein the step of receiving a signal that a new program is going to be executed is performed at the operating system kernel of the computing device(see col. 11, lines 7-12, col. 12, lines 52-63).

13. As per claim 4, McGee discloses wherein the step of determining whether the new program is the same as the allowed program(see col. 12, lines 52-63), computing a checksum for the new program, and comparing the new program's checksum to a checksum for the allowed program(col. 4, lines 35-49).

14. As per claim 5, McGee discloses wherein the step of determining whether the new program is the same as the allowed program comprises determining whether the new program has been modified (col. 9, lines 12-37).

15. As per claim 6, McGee discloses wherein the step of monitoring the new program while allowing it to execute includes controlling the files the program attempts to access during execution of the new program on the computing device(see col. 11, lines 3-36, col. 12, lines 19-

63).

16. As per claim 7, McGee discloses wherein the step of monitoring the new program while allowing it to execute includes controlling the registry settings the new program attempts to access during execution of the new program on the computing device(col. 6, lines 51-67, col. 12, lines 52-63).

17. As per claim 8, McGee discloses wherein the step of monitoring the new program while allowing it to execute includes controlling the network activity of the new program(col. 6, lines 51-67, col. 8, lines 11-40).

18. As per claim 9, McGee discloses wherein the step of monitoring the new program while allowing it to execute includes determining whether a user has previously approved operation of the new program(see col. 12, lines 19-63).

19. As per claim 10, McGee discloses the step of terminating the execution of the new program if the program performs suspicious activities(see col. 4, lines 40-61).

20. As per claim 11, McGee discloses the step of taking remedial measures to protect the security of other computing device components if the new program is not the same as the allowed program(col. 8, lines 11-40).

21. As per claim 12, McGee discloses a computer-readable medium having computer-executable instructions for performing the steps recited in claim 1(see col. 10, lines 23-38).

22. As per claim 13, McGee discloses a computer-implemented method for implementing security for a computing device(see col. 10, lines 23-38), interrupting the loading of a new program for operation with the computing device; validating the new program; if the new program is validated, permitting the new program to continue loading and to execute in

connection with the computing device. McGee does not disclose if the new program is not validated, monitoring the new program while it loads and executes in connection with the computing device. Bush discloses if the new program is not the same as the allowed program, monitoring the new program while allowing it to execute on the computing device[0007-0009]. It would have been obvious to one of ordinary skill in the art to include monitoring the new program while allowing it to execute on the computing device of Bush with McGee, because often there is a need for flexibility when dealing with untrusted code(i.e. new)[0009 of Bush]. Bush discloses there may be a need to permit untrusted code to have some defined access to the trusted computer resources, such as to particular data managed by the trusted code or to particular methods of the trusted code[0009 of Bush].

23. As per claim 14, McGee discloses wherein the step of interrupting the loading of a new program comprises: intercepting a signal from the computing device's operating system that the new program is loading, and suspending the loading of the new program(col. 8, lines 11-40).

24. As per claim 15, McGee discloses wherein the step of monitoring the new program includes intercepting a signal from the computing device's operating system((see col. 11, lines 7-12, col. 12, lines 52-63).

25. As per claim 16, McGee discloses wherein the step of validating the new program includes determining whether the new program corresponds with an approved program(col. 6, lines 19-50).

26. As per claim 17, McGee discloses wherein the step of validating the new program includes comparing a checksum for the new program with a previously determined checksum((col. 4, lines 35-49).

27. As per claim 18, McGee discloses wherein the step of validating the new program comprises analyzing characteristics of the new program(col. 8, lines 11-40).

28. As per claim 19, McGee discloses wherein the step of monitoring the new program includes controlling the files the new program attempts to access during execution of the new program(see col. 11, lines 3-36, col. 12, lines 19-63).

29. As per claim 20, McGee discloses a computer-readable medium having computer-executable instructions for performing the steps recited in claim 13(see col. 10, lines 23-38).

30. As per claim 21, McGee discloses a computer-implemented method for performing security for a computing device during a pre-execution phase(col. 10, lines 23-38) identifying an allowed program that is permitted to execute with the computing device(see col. 4, lines 1-23); receiving a signal that a new program is being loaded for execution with the computing device(see col. 11, lines 7-12, col. 12, lines 52-63); suspending the loading of the new program; comparing the new program to the allowed program(see col. 8, lines 11-40); and determining whether the new program is valid(col. 12, lines 52-63).

31. As per claim 23, McGee discloses wherein the step of comparing the new program to the allowed program includes comparing a checksum of the new program to a checksum of the allowed program(col. 4, lines 35-49).

32. As per claim 24, McGee discloses wherein the step of determining whether the new program is valid includes determining whether a checksum of the new program corresponds to a checksum of the allowed program(see col. 4, lines 35-49).

33. As per claim 25, McGee discloses a computer-readable medium having computer-executable instructions for performing the steps recited in claim 21(col. 10, lines 23-38).

34. As per claim 26, McGee discloses a computer-implemented method for performing security for a computing device during an execution phase(see col. 10, lines 23-38), intercepting a call to open a file from the computing device's operating system(see col. 11, lines 7-12); determining whether the file is restricted; if the file is restricted, taking remedial action to protect the computing device; and if the file is not restricted, permitting the call to open the file(see col. 8, lines 11-40).

35. As per claim 27, McGee discloses the step of receiving an instruction to monitor a call from an executing program(see col. 11, lines 7-12).

36. As per claim 28, McGee discloses the step of defining files that are restricted from access(see col. 11, lines 17-35).

37. As per claim 29, McGee discloses, wherein the step of defining files that are restricted includes designating a file that cannot be accessed in response to the call(see col. 11, lines 3-36).

38. As per claim 30, McGee discloses wherein the step of defining files that are restricted includes designating a file that cannot be modified in response to the call(see col. 11, lines 3-36, col. 13, lines 8-20).

39. As per claim 31, McGee discloses the step of alerting a security management module of the presence of a malicious program(see col. 13, lines 8-20).

40. As per claim 32, McGee discloses wherein the step of taking remedial action includes terminating the call(see col. 8, lines 11-40).

41. As per claim 33, McGee discloses wherein the step of taking remedial action includes alerting another security module of the existence of the call(see col. 8, lines 11-40, col. 13, lines 8-20).

42. As per claim 34, McGee discloses wherein the step of taking remedial action includes notifying the user of the existence of the call(see col. 8, lines 11-40, col. 11, lines 7-12).

43. As per claim 35, McGee discloses a computer-readable medium having computer-executable instructions for performing the steps recited in claim 26(see col. 10, lines 23-38).

44. Claims 3, 22 are rejected under 35 U.S.C. 103(a) as being unpatentable over McGee in view of Bush as applied to claim 1 and 21 respectively above, and further in view of Ioannidis.

45. As per claims 3, 22, McGee nor Bush discloses wherein the step of monitoring the new program while allowing it to execute is performed at the operating system kernel of the computing device. Ioannidis teaches wherein the step of monitoring the new program while allowing it to execute is performed at the operating system kernel of the computing device(see 3 The SubOS Architecture, and 3.1 Implementation, pg.3-4). It would have been obvious to one of ordinary skill in the art at the time of the invention to include the teachings of Ioannidis of monitoring the new program while allowing it to execute is performed at the operating system kernel of the computing device with McGee-Bush combination, allowing a new program to execute is performed at the operating system kernel, allows a new program to execute inside a safe interpreter that will limit its access to the file system(see 3 The SubOS Architecture).

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to JENISE E. JACKSON whose telephone number is (571)272-3791. The examiner can normally be reached on Increased Flex time, but generally in the office M-Fri(8-4:30)..

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kambiz Zand can be reached on (571) 272-3811. 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.

January 14, 2009
/J. E. J./
Examiner, Art Unit 2439

/Kambiz Zand/
Supervisory Patent Examiner, Art Unit 2434

Notice of References Cited	Application/Control No. 10/336,299	Applicant(s)/Patent Under Reexamination SATTERLEE ET AL.	
	Examiner JENISE E. JACKSON	Art Unit 2439	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-			
	B	US-			
	C	US-			
	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	Ioannidis, Sub-Operating Systems: A New Approach to Application Security, 2000, pg. 110-115.
	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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1319

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10336299	SATTERLEE ET AL.
	Examiner	Art Unit
	JENISE E JACKSON	2439

✓	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	07/18/2008	01/14/2009							
	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	✓	✓							
	32	✓	✓							
	33	✓	✓							
	34	✓	✓							
	35	✓	✓							
	36	✓	✓							

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1320

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10336299	SATTERLEE ET AL.
	Examiner	Art Unit
	JENISE E JACKSON	2439

✓	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	07/18/2008	01/14/2009							
	37	✓	✓							
	38	✓	✓							
	39	✓	✓							
	40	✓	✓							

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1321

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10336299	SATTERLEE ET AL.
	Examiner	Art Unit
	JENISE E JACKSON	2439

SEARCHED			
Class	Subclass	Date	Examiner
713	164-165, 167	1/14/08	JJ
726	22, 24, 26	1/14/08	JJ

SEARCH NOTES		
Search Notes	Date	Examiner
Updated West text search	1/14/08	JJ
Updated West class and text search	1/14/08	JJ
Updated Inventor and Assignee search	1/14/08	JJ
Consulted Matt Heneghan	7/15/08	JJ

INTERFERENCE SEARCH			
Class	Subclass	Date	Examiner

WEST Search History for Application 10336299

Creation Date: 2009011410:02

Query	DB	Op.	Plur.	Thes.	Date
(allowed or approved)with(program or code or software or executable or application)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
(determin$4)near5(new)with(program or software or code or executable or application)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
(pre-approv$ or approv$4 or allow$4)with(list or database) and ((determin$4)near5(new)with(program or software or code or executable or application))	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
((pre-approv$ or approv$4 or allow$4)with(list or database) and (determin$4)near5(new)with(program or software or code or executable or application)) and sandbox	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
Monitor$4 near5(kernel)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
(validat$4 or authenticat$4 or authoriz$4 or verif$4)near5(new)near4(program or application or software or code or executable)with(approv$4 or allow$4)near5(program or application or code or software or executable)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
(compar$4)near5(new)with(program or application or software or code or executable)near5(list or database)with(approv$4 or allow$4 or pre-approv$4)	PGPB, USPT, EPAB, JPAB,	OR	YES		07-18-2008

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1323

	DWPI, TDBD				
(monitor$4)near5(new)with(program or software or application or code or executable) and (allow$4 or approv$4)near5(computer)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
((monitor$4)near5(new)with(program or software or application or code or executable) and (allow$4 or approv$4)near5(computer)) and sandbox	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
(execut$4)near5(new)with(program or software or executable or code or application)near5(sandbox)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
TPM and sandbox	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
(compar$4)with(new adj program)with(approv$4 or allow$4)near4(program)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
(list)with(approv$4 or allow$4)near5(program)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
((list)with(approv$4 or allow$4)near5(program)) and sandbox	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
		OR	YES		07-18-2008

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1324

(new)near5(program or application or software or code or executable)same(allow$4 or approv$4)near5(program or application or software or code or executable)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD				
((new)near5(program or application or software or code or executable)same(allow$4 or approv$4)near5(program or application or software or code or executable)) and sandbox	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
726/1.ccls.and (pre-approv? or approv? or allow?)near5(program or executable or software)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
726/2.ccls.and (pre-approv? or approv? or allow?)near5(program or executable or software)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
726/14.ccls.and (pre-approv? or approv? or allow?)near5(program or executable or software)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
726/22.ccls.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
726/24.ccls.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
726/26.ccls.	PGPB, USPT, EPAB, JPAB,	OR	YES		07-18-2008

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1325

	DWPI, TDBD				
713/161.ccls.and (pre-approv? or approv? or allow?)near5(program or executable or software)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
713/164.ccls.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
713/165.ccls.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
713/182.ccls.and (pre-approv? or approv? or allow?)near5(program or executable or software)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
713/187.ccls.and (pre-approv? or approv? or allow?)near5(program or executable or software)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
713/188.ccls.and (pre-approv? or approv? or allow?)near5(program or executable or software)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
713/167.ccls.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
Thomas near James near Satterlee.in.		OR	YES		07-18-2008

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1326

	PGPB, USPT, EPAB, JPAB, DWPI, TDBD				
William near Frank near Hackenberger.in.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
Internet near Security near Systems.as. and (execut?)near4(program)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
TPM and sandbox	PGPB	OR			07-18-2008
sandbox	PGPB	OR			07-18-2008
(allow$4)near5(new)with(program or application or executable or software or code)with(run$4 or execut$4) and (sandbox)	PGPB	OR			07-18-2008
(sandbox) and (monitor$4)with(new)with(program or software or code or executable or application)	PGPB	OR			07-18-2008
(sandbox) and Gong.in.	PGPB	OR			07-18-2008
(sandbox) and (allow$4)with(execut$4)near5(computer)	PGPB	OR			07-18-2008
(sandbox) and (applet)	PGPB	OR			07-18-2008
(allowed or approv$4 or authoriz$4)near5(program or software or application or executable or code)near5(run$4 or execut$4)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR			07-18-2008
(new)near5(program or executable or software or application or code)near5(compar$4 or validat$4 or authenticat$4) and ((allowed or approv$4 or authoriz$4)near5(program or software or application or executable or code)near5(run$4 or execut$4))	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR			07-18-2008
((new)near5(program or executable or software or application or code)near5(compar$4 or validat$4 or	PGPB, USPT,	OR			07-18-2008

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1327

authenticat$4) and (allowed or approv$4 or authoriz$4)near5(program or software or application or executable or code)near5(run$4 or execut$4)) and sandbox	EPAB, JPAB, DWPI, TDBD				
(new)near5(program or executable or software or application or code)near5(compar$4 or validat$4 or authenticat$4)with(allow$4 or approv$4)near5(program or software or application or executable or code)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR			07-18-2008
(new or unknown)with(program or software or application or code or applet or activex)near5(non-allow$4 or non-approv$4)with(program or software or application or code or applet or activex)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR			07-18-2008
((allowed or approv$4 or authoriz$4)near5(program or software or application or executable or code)near5(run$4 or execut$4)) and (sandbox)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR			07-18-2008
(new or unknown)near5(no near4 validat$4 or no near4 authoriz$4 or no near5 authenticat$4)with(program or software or application or code) and ((allow$4)near5(new)with(program or application or executable or software or code)with(run$4 or execut$4) and sandbox)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR			07-18-2008
(new or unknown)near5(no near4 validat$4 or no near4 authoriz$4 or no near5 authenticat$4)with(program or software or application or code) and ((allowed or approv$4 or authoriz$4)near5(program or software or application or executable or code)near5(run$4 or execut$4) and sandbox)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR			07-18-2008
(execut$4 or run$4)near5(new)with(program or software or executable or code or application)near5(kernel)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
(monitor$4)near5(new)with(program or software or code application)near5(execut$4 or run$4)with(kernel)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1328

(new)near5(program or executable or software or application or code)near5(compar$4 or validat$4 or authenticat$4)and(allowed or approv$4 or authoriz$4)near5(program or software or application or executable or code)near5(run$4 or execut$4)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
(allow$4 or approv$4 or authoriz$4)near5(program or software or application or code)with(execut$4 or run$4)near5(computer or pc or computing near device or workstation)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
(new or unknown)with(program or software or code or application)near5(execut$4 or run$4)with(computer or pc or computing near device or workstation)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
(compar$4 or validat$4 or authenticat$4 or verif$5)near5(new or unknown)with(program or software or code or application)near5(allow$4 or approv$4 or authoriz$4)with(program or software or application or code) and ((new or unknown)with(program or software or code or application)near5(execut$4 or run$4)with(computer or pc or computing near device or workstation))	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
726/1.ccls. and (pre-approv$3 or approv$3 or allow$3)near5(program or software or code or application)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
726/2.ccls.and (pre-approv$3 or approv$3 or allow$3)near5(program or software or code or application)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
726/22.ccls.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
726/24.ccls.	PGPB, USPT, EPAB,	OR	YES		01-14-2009

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1329

	JPAB, DWPI, TDBD				
726/26.ccls.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
713/161.ccls.and (pre-approv$3 or approv$3 or allow$3)near5(program or code or software or application)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
726/14.ccls.and (pre-approv$3 or approv$3 or allow$3)near5(program or code or software or application)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
713/161.ccls. and (pre-approv$4 or approv$4 or allow$3)near5(program or software or application or code)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
713/164.ccls.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
713/165.ccls.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
713/182.ccls.and (pre-approv$4 or approv$4 or allow$3)near5(program or software or application or code)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1330

713/187.ccls.and (pre-approv$4 or approv$4 or allow$3)near5(program or software or application or code)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
713/188.ccls.and (pre-approv$4 or approv$4 or allow$3)near5(program or software or application or code)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
713/167.ccls.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
Thomas near James near Satterlee.in.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
William near Frank near Hackenberger.in.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
Internet near Security near Systems.as. and (execut$4)near4(program)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1331

PATENTS

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	)	
	)	Attorney Docket No. **05456.105038**
Thomas James Satterlee et al.	)	
	)	Group Art Unit: **2439**
Application No. **10/336,299**	)	
	)	Confirmation No.: **7850**
Filed: **January 3, 2003**	)	
	)	Examiner: **Jackson, Jenise E.**
For: **System and Method for the Managed Security Control of Processes on a Computer**	)	

APPEAL BRIEF

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

Sir:

Further to the Notice of Appeal filed concurrently herewith in the above referenced application, Appellants hereby submit this brief under 37 C.F.R. § 1.191 to appeal the Examiner's rejection of this application, as reported in the Office Action mailed on January 16, 2009.

Please apply the $500.00 appeal brief fee paid with the previous appeal brief submitted September 4, 2007 and the $10 additional fee paid with the previous appeal brief submitted May 9, 2008.

The Commissioner is authorized to charge any additional fee required for this Appeal Brief, or to credit any overpayment, to Deposit Account No. 09-0457.

Respectfully submitted,

/ Robert T. Neufeld /

Robert T. Neufeld
Reg. No. 48,394

I hereby certify that this correspondence is being electronically transmitted via EFS-web to the Commissioner for Patents on April 16, 2009.

/ Robert T. Neufeld /
Robert T. Neufeld, Reg. No. 48,394

Table of Contents

Real Party in Interest 3

Related Appeals and Interferences 4

Status of Claims 5

Status of Amendments 6

Summary of Claimed Subject Matter 7

Grounds of Rejection to be Reviewed on Appeal 8

Argument 9

Conclusion 12

APPENDIX 1 13

APPENDIX 2 14

APPENDIX 3 15

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1333

Real Party in Interest

The real party in interest is IBM Internet Security Services, formerly operating as Internet Security Systems, Inc. IBM Internet Security Services is a wholly owned subsidiary of International Business Machines Corporation.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1334

Related Appeals and Interferences

None.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1335

Status of Claims

Claims 1-40 stand rejected in the most recent Office Action of January 16, 2009.

The rejection of Claim 3 is appealed.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1336

Status of Amendments

No amendments have been filed subsequent to the non-final rejection mailed on January 16, 2009.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1337

Summary of Claimed Subject Matter

The method of dependent Claim 3 includes the following steps from Claims 1 and 3: (1) identifying an allowed program that is permitted to execute on the computing device (page 11, lines 30 to 31; page 12, lines 18 to 22; step 300); (2) receiving a signal that a new program is going to be executed on the computing device (page 5, lines 14 to 16; page 13, lines 26 to 27; step 410); (3) suspending the execution of the new program on the computing device (page 14, lines 7 to 8; page 6, lines 1 to 3; step 430); (4) determining whether the new program is the same as the allowed program (page 5, lines 16 to 17; page 14, lines 12 to 15; step 435); (5) if the new program is the same as the allowed program, permitting the new program to execute on the computing device (page 5, lines 17 to 19; page 14, lines 15 to 17; step 440); (6) if the new program is not the same as the allowed program, monitoring the new program while allowing it to execute on the computing device (page 5, lines 21 to 23; page 14 lines 17 to 20; step 600); and (7) wherein the step of monitoring the new program while allowing it to execute is performed at the operating system kernel of the computing device (page 9, lines 7-16; page 10, lines 14-17; page 11, lines 1-3; steps 410 and 645).

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1338

Grounds of Rejection to be Reviewed on Appeal

The following issue is presented on appeal:

(1) Whether, under 35 U.S.C. § 103, Claim 3 is obvious in view of the combination of U.S. Patent No. 6,694,434 to McGee ("McGee"), U.S. Published Patent Application No. 2002/0184520 to Bush ("Bush"), and the article entitled "Sub-Operating Systems: A New Approach to Application Security" by Sotiris Ioannidis, et al. ("Ioannidis").

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1339

Argument

Rejection of claims under 35 U.S.C. § 103(a)

Applicants submit that the prior art of record does not render pending Claim 3 obvious.

The Legal Standard for 35 U.S.C. § 103

The U.S. Patent and Trademark Office has the burden under 35 U.S.C. § 103 to establish a *prima facie* case of obviousness. *In re Warner et al.*, 379 F.2d 1011, 154 U.S.P.Q. 173, 177 (C.C.P.A. 1967), *In re Fine,* 837 F.2d 1071, 1074, 5 U.S.P.Q.2d 1596, 1598-99 (Fed. Cir. 1988). To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art and not based on applicants' disclosure. *In re Vaeck, 947 F.2d 488, 20 U.S.P.Q.2d 1438 (Fed. Cir. 1991).* The references cited by the Examiner do not meet all three criteria.

While the "teaching, suggestion, motivation test" should not be applied rigidly, it nonetheless provides insight into the obviousness inquiry. *KSR Int'l v. Teleflex Inc.*, 82 U.S.P.Q.2d 1385, 1391, 1396 (2007). The prior art must provide one of ordinary skill in the art with the motivation to make the proposed modification needed to arrive at the claimed invention. *In re Geiger*, 815 F.2d 686, 2 U.S.P.Q.2d 1276 (Fed. Cir. 1987); *in re Lalu and Foulletier*, 747 F.2d 703, 705, 223 U.S.P.Q. 1257, 1258 (Fed. Cir. 1984). Claims for an invention are not *prima facie* obvious if the primary references do not suggest all elements of the claimed invention and the prior art does not suggest the modifications that would bring the primary references into conformity with the application claims. *In re Fritch*, 23 U.S.P.Q.2d, 1780 (Fed. Cir. 1992). *In re Laskowski*, 871 F.2d 115 (Fed. Cir. 1989). This is not possible when the claimed invention achieves more than what any or all of the prior art reference allegedly suggest, expressly or by reasonable implication.

It is clear that to establish a rejection under 35 U.S.C. § 103 the cited references must (1) recite each element of the claims, (2) provide one of skill in the art with the motivation to combine the cited reference as applications have done and (3) provide one of ordinary skill in the

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1340

art with a reasonable expectation of success. The references cited by the Examiner clearly do not meet all three criteria and the current rejection of the claims-in-issue lack proper support.

Analysis

The McGee, Bush and Ioannidis references fail to disclose or render obvious under 35 U.S.C. § 103 the recitations that are present in Claim 3, which depends from Claim 1.

Dependent Claim 3

In the Office Action, the Examiner rejected independent Claim 1 as obvious in view of the combination of the McGee and Bush references. The Examiner further rejected dependent Claim 3 as obvious in view of the disclosures in Ioannidis.

In the Office Action, the Examiner acknowledges that McGee does not disclose the last step of Claim 1 where a program is not pre-approved and is therefore monitored while it is allowed to execute. The Examiner asserts that Bush discloses the last step of Claim 1 in that paragraphs 0007 - 0009 of Bush describe a "sandbox method" where trusted code is permitted to execute with full access to a computer system's resources and untrusted code is permitted to execute, but only with limited access to a computer system's resources. The Bush reference goes on to describe using a virtual machine, which is separate from the operating system, to process untrusted code. (See paragraphs 0010 and 0012.)

Claim 3, which depends from Claim 1, provides further detail concerning the last step of Claim 1 in that the monitoring of the new program while allowing it to execute "is performed at the operating system kernel of the computing device." The Examiner contends that Sections 3 and 3.1 on pages 3-4 of Ioannidis disclose the step recited in dependent Claim 3.

As an initial matter, Applicants note that there is no publication date on Ioannidis and it is unclear when the article was published. Therefore, Applicants submit that Ioannidis is not prior art to the claims pending in the present application and cannot be applied to reject the pending claims.

Furthermore, Applicants respectfully submit that Ioannidis does not disclose monitoring an unapproved program from the operating system kernel while allowing the program to execute. While Ioannidis does describe its solution as being implemented at the kernel, Ioannidis is not performing the step of monitoring at the operating system kernel, as recited in Claim 3. Instead,

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1341

Ioannidis describes a solution where a restricted set of permissions are assigned to an object. Applicants submit that assigning a restricted set of permissions to an object is not the same as monitoring the execution of a program at the operating system kernel.

Furthermore, Applicants submit that the disclosures in Bush of a "sandbox method" and using a virtual machine do not teach the method recited in Claim 3. In particular, Claim 3 recites "monitoring the new program at an operating system kernel of the computing device". Allowing the program to execute and monitoring the new program at the operating system kernel are distinct from the sandbox or virtual machine methods disclosed in Bush where the untrusted code is processed in an artificial environment separate from the trusted code. As explained in the Background portion of the specification of the present application, "[v]irtual execution is limited in its ability to detect harmful code because it does not actually execute every process of the code." (See page 2, line 30 - page 3, line 1.) In contrast to the virtual execution of untrusted code described in Bush, Claim 3 recites allowing the unknown program to execute, but monitoring the execution at the operating system kernel.

Applicants also note that their positions are consistent with the Notice of Allowance recently issued in the related application assigned Serial Number 11/824,486. Therefore, Applicants submit that Claim 3 is patentably distinct from the art of record.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1342

Conclusion

In view of the arguments presented herein, Applicants respectfully request that the rejection in this matter be vacated, and that this application be returned to the Examiner with instructions to enter a notice of allowance.

Respectfully submitted,

/ Robert T. Neufeld /

Robert T. Neufeld
Reg. No. 48,394

KING & SPALDING LLP
1180 Peachtree Street
34th Floor
Atlanta, GA 30309
(404) 572-4600 (Telephone)
(404) 572-5134 (Facsimile)

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1343

APPENDIX 1

CLAIMS APPENDIX

1. (Original) A computer-implemented method for implementing security for a computing device, comprising the steps of:

identifying an allowed program that is permitted to execute on the computing device;

receiving a signal that a new program is going to be executed on the computing device;

suspending the execution of the new program on the computing device;

determining whether the new program is the same as the allowed program;

if the new program is the same as the allowed program, permitting the new program to execute on the computing device; and

if the new program is not the same as the allowed program, monitoring the new program while allowing it to execute on the computing device.

3. (Original) The method of Claim 1, wherein the step of monitoring the new program while allowing it to execute is performed at the operating system kernel of the computing device.

The rejection of Claims 1, 2 and 4-40 are not appealed at this time.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1344

APPENDIX 2

EVIDENCE APPENDIX

None.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1345

APPENDIX 3

RELATED PROCEEDINGS APPENDIX

None.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1346

Electronic Acknowledgement Receipt

EFS ID:	5166298
Application Number:	10336299
International Application Number:	
Confirmation Number:	7850
Title of Invention:	System and method for the managed security control of processes on a computer system
First Named Inventor/Applicant Name:	Thomas James Satterlee
Customer Number:	69151
Filer:	Robert Timothy Neufeld
Filer Authorized By:	
Attorney Docket Number:	05456.105038
Receipt Date:	16-APR-2009
Filing Date:	03-JAN-2003
Time Stamp:	14:50:55
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Notice of Appeal Filed	END9_2007_5119US2_Notice_Appeal_4_16_09.pdf	77370 1248a50a48ce6b9bcc612e6229b25a5850949fb4	no	1

Warnings:	
Information:	

2	Appeal Brief Filed	END9_2007_5119US2_Appeal_Brief_4_16_09.pdf	127801 cdf3fbb84d58750dbc02f94314c681a1ed7d1b4d	no	15

Warnings:	
Information:	
Total Files Size (in bytes):	205171

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.

PATENTS

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	)	
	)	Atty Docket: **05456.105038**
Thomas James Satterlee et al.	)	
	)	Art Unit: **2439**
Application No.: **10/336,299**	)	
	)	Examiner: **Jackson, Jenise E.**
Filing Date: **January 3, 2003**	)	
	)	Confirmation No.: **7850**
Title: **System and Method for the Managed Security Control of Processes of a Computer**	)	

NOTICE OF APPEAL

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

Sir:

Applicants hereby appeal to the Board of Patent Appeals and Interferences from the last decision of the Examiner mailed January 16, 2009.

Please apply the $500 Notice of Appeal fee previously paid with the notice submitted on July 2, 2007 and the $10 additional fee previously paid with the notice submitted on March 10, 2008.

The Commissioner is authorized to charge any additional fee required for this Notice of Appeal, or to credit any overpayment, to Deposit Account No. 09-0475. A duplicate of this paper is enclosed.

Respectfully submitted,

/ Robert T. Neufeld /

KING & SPALDING LLP
1180 Peachtree Street, N.E., 34th Floor
Atlanta, Georgia 30309-3521
(404) 572-4600

Robert T. Neufeld
Attorney for Applicants
Reg. No. 48,394

I hereby certify that this correspondence is being electronically transmitted via EFS-Web to the Commissioner for Patents on April 16, 2009.

/ Robert T. Neufeld /
Robert T. Neufeld, Reg. No. 48,394

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

NOTICE OF ALLOWANCE AND FEE(S) DUE

69151 7590 07/24/2009

KING & SPALDING, LLP
INTELLECTUAL PROPERTY DEPT. - PATENTS
1180 PEACHTREE STREET, N.E.
ATLANTA, GA 30309-3521

EXAMINER	
JACKSON, JENISE E	
ART UNIT	PAPER NUMBER
2439	

DATE MAILED: 07/24/2009

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/336,299	01/03/2003	Thomas James Satterlee	05456.105038	7850

TITLE OF INVENTION: SYSTEM AND METHOD FOR THE MANAGED SECURITY CONTROL OF PROCESSES ON A COMPUTER SYSTEM

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1510	$300	$0	$1810	10/26/2009

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the SMALL ENTITY status shown above.

If the SMALL ENTITY is shown as YES, verify your current SMALL ENTITY status:

A. If the status is the same, pay the TOTAL FEE(S) DUE shown above.

B. If the status above is to be removed, check box 5b on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and twice the amount of the ISSUE FEE shown above, or

If the SMALL ENTITY is shown as NO:

A. Pay TOTAL FEE(S) DUE shown above, or

B. If applicant claimed SMALL ENTITY status before, or is now claiming SMALL ENTITY status, check box 5a on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and 1/2 the ISSUE FEE shown above.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

PTOL-85 (Rev. 08/07) Approved for use through 08/31/2010.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1350

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

69151 7590 07/24/2009

KING & SPALDING, LLP
INTELLECTUAL PROPERTY DEPT. - PATENTS
1180 PEACHTREE STREET, N.E.
ATLANTA, GA 30309-3521

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)

(Signature)

(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/336,299	01/03/2003	Thomas James Satterlee	05456.105038	7850

TITLE OF INVENTION: SYSTEM AND METHOD FOR THE MANAGED SECURITY CONTROL OF PROCESSES ON A COMPUTER SYSTEM

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1510	$300	$0	$1810	10/26/2009

EXAMINER	ART UNIT	CLASS-SUBCLASS
JACKSON, JENISE E	2439	713-164000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).

❑ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.

❑ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list

(1) the names of up to 3 registered patent attorneys or agents OR, alternatively,

(2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1 _____

2 _____

3 _____

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE (B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent) : ❑ Individual ❑ Corporation or other private group entity ❑ Government

4a. The following fee(s) are submitted:

❑ Issue Fee

❑ Publication Fee (No small entity discount permitted)

❑ Advance Order - # of Copies _____

4b. Payment of Fee(s): **(Please first reapply any previously paid issue fee shown above)**

❑ A check is enclosed.

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

❑ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment, to Deposit Account Number _____ (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)

❑ a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27.

❑ b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature _____ Date _____

Typed or printed name _____ Registration No. _____

This collection of information is required by 37 CFR 1.311. 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, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1351

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.
10/336,299	01/03/2003	Thomas James Satterlee	05456.105038	7850

69151 7590 07/24/2009

KING & SPALDING, LLP
INTELLECTUAL PROPERTY DEPT. - PATENTS
1180 PEACHTREE STREET, N.E.
ATLANTA, GA 30309-3521

EXAMINER	
JACKSON, JENISE E	
ART UNIT	PAPER NUMBER
2439	

DATE MAILED: 07/24/2009

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment to date is 952 day(s). If the issue fee is paid on the date that is three months after the mailing date of this notice and the patent issues on the Tuesday before the date that is 28 weeks (six and a half months) after the mailing date of this notice, the Patent Term Adjustment will be 952 day(s).

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

PTOL-85 (Rev. 08/07) Approved for use through 08/31/2010.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1352

Notice of Allowability	Application No.	Applicant(s)	
	10/336,299	SATTERLEE ET AL.	
	Examiner	Art Unit	
	JENISE E. JACKSON	2439	

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

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *4/16/09*.

2. ☒ The allowed claim(s) is/are *1-2, 4-25 and 36-40*.

3. ☐ 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 the:
      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)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

4. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.

5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
   (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
      1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____.
   (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☒ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
3. ☐ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date _____
4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. ☐ Notice of Informal Patent Application
6. ☒ Interview Summary (PTO-413), Paper No./Mail Date *20090714* .
7. ☒ Examiner's Amendment/Comment
8. ☒ Examiner's Statement of Reasons for Allowance
9. ☐ Other _____.

/Christian LaForgia/
Primary Examiner, Art Unit 2439

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1353

Interview Summary	Application No.	Applicant(s)	
	10/336,299	SATTERLEE ET AL.	
	Examiner	Art Unit	
	JENISE E. JACKSON	2439	

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

(1) *JENISE E. JACKSON*. (3)_____.

(2) *Robert T. Neufeld*. (4)_____.

Date of Interview: *7/10/09, 7/13/09*.

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: *1,3,13,21 and 26-36*.

Identification of prior art discussed: _____.

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: *The Examiner contacted the Attorney of record, Robert Neufeld, to discuss incorporating dependent claim 3 into independent claims 1, 13, 21, and 36. The Examiner and Mr. Neufeld also discussed cancelling claims 26-35, and agreement was reached and Mr. Neufeld gave the Examiner authorization to cancel claims 26-35*.

(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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1354

EXAMINER'S AMENDMENT

1. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Robert T. Neufeld on July 13, 2009.

The application has been amended as follows:

1. (Currently Amended) A computer-implemented method for implementing security for a computing device, comprising the steps of:

identifying an allowed program that is permitted to execute on the computing device;

receiving a signal that a new program is going to be executed on the computing device;

suspending the execution of the new program on the computing device;

determining whether the new program is the same as the allowed program;

if the new program is the same as the allowed program, permitting the new program to execute on the computing device; and

if the new program is not the same as the allowed program, monitoring the new program while allowing it to execute on the computing device, **wherein the step of monitoring the new program while allowing it to execute is performed at the operating system kernel of the computing device.**

3. **(Canceled)**

13. (Currently Amended) A computer implemented method for implementing security for a computing device comprising the steps of:

interrupting the loading of a new program for operation with the computing device;

validating the new program;

if the new program is validated, permitting the new program to continue loading and to execute in connection with the computing device;

if the new program is not validated, monitoring the new program while it loads and executes in connection with the computing device, **wherein the step of monitoring the new program while it executes is performed at the operating system kernel of the computing device.**

21. (Currently Amended) A computer-implemented method for performing security for a computing device during a pre-execution phase comprising the steps of:

identifying an allowed program that is permitted to execute with the computing device;

receiving a signal that a new program is being loaded for execution with the computing device;

suspending the loading of the new program;

comparing the new program to the allowed program; and

determining whether the new program is valid;

if the new program is valid, permitting the new program to execute on the computing device; and

if the new program is not valid, monitoring the new program while allowing it to execute on the computing device, wherein the step of monitoring the new program while allowing it to execute is performed at the operating system kernel of the computing device.

26. **Canceled**

27. **Canceled**

28. **Canceled**

29. **Canceled**

30. **Canceled**

31. **Canceled**

32. **Canceled**

33. **Canceled**

34. **Canceled**

35. **Canceled**

36. (Currently Amended) A system for managing security of a computing device comprising:

a pre-execution module operable for receiving notice from the computing device's operating system that a program is being loaded onto the computing device;

a validation module coupled to the pre-execution monitor operable for determining whether the program is valid;

a detection module coupled to the pre-execution monitor operable for intercepting a trigger from the computing device's operating system; and an execution module coupled to the

detection module and operable for monitoring, **at the operating system kernel of the computing device**, the program in response to the trigger intercepted by the detection module.

Reasons for Allowance

1. Claims 1-2, 4-25 and 36-40 are allowed. The following is an examiner's statement of reasons for allowance: see reasons listed below:

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

2. Prior art fails to disclose or suggest, "if the new program is not the same as the allowed program, monitoring the new program while allowing it to execute on the computing device, wherein the step of monitoring the new program while allowing it to execute is performed at the operating system kernel of the computing device", and example of prior art that fails to disclose or suggest these limitations is Barber et al. Barber et al. discloses a process A and B in which are invoked each time that an application program is launched. Process A and B include attributes that are maintained within a kernel and include credentials/privileges. Thus, if process A or B are not validated by there credentials/privileges process A and B will not execute. The prior art of Barber is in contrast to the claims, that disclose, "if the new program is not the same as the allowed program, monitoring the new program while allowing it to execute on the computing device, wherein the step of monitoring the new program while allowing it to execute is performed at the operating system kernel of the computing device".

3. Prior art fails to disclose or suggest, "if the new program is not the same as the allowed program, monitoring the new program while allowing it to execute on the computing device, wherein the step of monitoring the new program while allowing it to execute is performed at the operating system kernel of the computing device", an example of prior art that fails to disclose or suggest these limitations is Necula et al. Necula discloses a code consumer defines a safety policy. The safety policy is a set of rules that specifies precisely under what conditions the execution of the untrusted code is considered to be safe. The code consumer extracts a safety predicate from the untrusted code. The validity of the safety predicate guarantees the execution of the code will not violate the safety policy. If it is valid, than the untrusted code is installed and executed, if it is found to be invalid the untrusted code is discarded. Necula is in direct contrast to claims that disclose, if the new program is not the same as the allowed program, monitoring the new program while allowing it to execute on the computing device, wherein the step of monitoring the new program while allowing it to execute is performed at the operating system kernel of the computing device".

4. Prior art fails to disclose or suggest, "if the new program is not the same as the allowed program, monitoring the new program while allowing it to execute on the computing device, wherein the step of monitoring the new program while allowing it to execute is performed at the operating system kernel of the computing device", an example of prior art that fails to disclose or suggest these limitations is Bush. Bush discloses a sandbox method where the trusted code is permitted to execute with full access to a computer system's resources and untrusted code is permitted to execute, but only with limited access to a computer system's resources. The Bush

reference discloses a virtual machine, which is separate from the operating system to process untrusted code.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to JENISE E. JACKSON whose telephone number is (571)272-3791. The examiner can normally be reached on Increased Flex time, but generally in the office M-Fri(8-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Edan Orgad can be reached on (571) 272-7884. 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.

/Christian LaForgia/
Primary Examiner, Art Unit 2439

July 14, 2009
/J. E. J./

Examiner, Art Unit 2439

Notice of References Cited	Application/Control No. 10/336,299	Applicant(s)/Patent Under Reexamination SATTERLEE ET AL.	
	Examiner JENISE E. JACKSON	Art Unit 2439	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-6,128,774	10-2000	Necula et al.	717/146
*	B	US-7,398,532	07-2008	Barber et al.	719/328
*	C	US-5,272,754	12-1993	Boerbert, William E.	713/159
*	D	US-5,359,713	10-1994	Moran et al.	710/52
	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.

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1362

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10336299	SATTERLEE ET AL.
	Examiner	**Art Unit**
	JENISE E JACKSON	2439

✓	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	07/18/2008	01/14/2009	07/14/2009						
14	1	✓	✓	=						
15	2	✓	✓	=						
	3	✓	✓	-						
16	4	✓	✓	=						
17	5	✓	✓	=						
18	6	✓	✓	=						
19	7	✓	✓	=						
20	8	✓	✓	=						
21	9	✓	✓	=						
22	10	✓	✓	=						
23	11	✓	✓	=						
24	12	✓	✓	=						
6	13	✓	✓	=						
7	14	✓	✓	=						
8	15	✓	✓	=						
9	16	✓	✓	=						
10	17	✓	✓	=						
11	18	✓	✓	=						
12	19	✓	✓	=						
13	20	✓	✓	=						
25	21	✓	✓	=						
26	22	✓	✓	=						
27	23	✓	✓	=						
28	24	✓	✓	=						
29	25	✓	✓	=						
	26	✓	✓	-						
	27	✓	✓	-						
	28	✓	✓	-						
	29	✓	✓	-						
	30	✓	✓	-						
	31	✓	✓	-						
	32	✓	✓	-						
	33	✓	✓	-						
	34	✓	✓	-						
	35	✓	✓	-						
1	36	✓	✓	=						

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1363

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10336299	SATTERLEE ET AL.
	Examiner	Art Unit
	JENISE E JACKSON	2439

✓	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	07/18/2008	01/14/2009	07/14/2009						
2	37	✓	✓	=						
3	38	✓	✓	=						
4	39	✓	✓	=						
5	40	✓	✓	=						

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1364

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10336299	SATTERLEE ET AL.
	Examiner	**Art Unit**
	JENISE E JACKSON	2439

ORIGINAL		INTERNATIONAL CLASSIFICATION									
CLASS	SUBCLASS	CLAIMED					NON-CLAIMED				
713	164	H	0	4	L	29 / 06 (2006.01.01)	G	0	6	F	7 / 04 (2006.01.01)

CROSS REFERENCE(S)					
CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)				
713	161	165	167	182	187
713	188				
726	1	2	14	22	24
726	26				

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

Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
14	1	10	17		33										
15	2	11	18		34										
	3	12	19		35										
16	4	13	20	1	36										
17	5	25	21	2	37										
18	6	26	22	3	38										
19	7	27	23	4	39										
20	8	28	24	5	40										
21	9	29	25												
22	10		26												
23	11		27												
24	12		28												
6	13		29												
7	14		30												
8	15		31												
9	16		32												

/JENISE E JACKSON/ Examiner.Art Unit 2439	7/14/09	**Total Claims Allowed:**	
(Assistant Examiner)	(Date)	29	
/Christian LaForgia/ Primary Examiner.Art Unit 2439	07/14/2009	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner)	(Date)	21	1

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1365

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10336299	SATTERLEE ET AL.
	Examiner	Art Unit
	JENISE E JACKSON	2439

SEARCHED			
Class	Subclass	Date	Examiner
713	164-165, 167	7/14/09	JJ
726	22, 24, 26	7/14/09	JJ

SEARCH NOTES		
Search Notes	Date	Examiner
Updated West text search	7/14/09	JJ
Updated West class and text search	7/14/09	JJ
Updated Inventor and Assignee search	7/14/09	JJ
Consulted Christian Laforgia	7/13/09	JJ
Non-patent literature search	7/9/09	JJ

INTERFERENCE SEARCH			
Class	Subclass	Date	Examiner
See West Interference Search .CLM.		7/14/09	JJ

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1366

WEST Search History for Application 10336299

Creation Date: 2009071412:08

Query	DB	Op.	Plur.	Thes.	Date
(allowed or approved)with(program or code or software or executable or application)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
(determin$4)near5(new)with(program or software or code or executable or application)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
(pre-approv$ or approv$4 or allow$4)with(list or database) and ((determin$4)near5(new)with(program or software or code or executable or application))	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
((pre-approv$ or approv$4 or allow$4)with(list or database) and (determin$4)near5(new)with(program or software or code or executable or application)) and sandbox	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
Monitor$4 near5(kernel)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
(validat$4 or authenticat$4 or authoriz$4 or verif$4)near5(new)near4(program or application or software or code or executable)with(approv$4 or allow$4)near5(program or application or code or software or executable)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
(compar$4)near5(new)with(program or application or software or code or executable)near5(list or database)with(approv$4 or allow$4 or pre-approv$4)	PGPB, USPT, EPAB, JPAB,	OR	YES		07-18-2008

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1367

	DWPI, TDBD				
(monitor$4)near5(new)with(program or software or application or code or executable) and (allow$4 or approv$4)near5(computer)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
((monitor$4)near5(new)with(program or software or application or code or executable) and (allow$4 or approv$4)near5(computer)) and sandbox	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
(execut$4)near5(new)with(program or software or executable or code or application)near5(sandbox)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
TPM and sandbox	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
(compar$4)with(new adj program)with(approv$4 or allow$4)near4(program)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
(list)with(approv$4 or allow$4)near5(program)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
((list)with(approv$4 or allow$4)near5(program)) and sandbox	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
		OR	YES		07-18-2008

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1368

(new)near5(program or application or software or code or executable)same(allow$4 or approv$4)near5(program or application or software or code or executable)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD				
((new)near5(program or application or software or code or executable)same(allow$4 or approv$4)near5(program or application or software or code or executable)) and sandbox	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
726/1.ccls.and (pre-approv? or approv? or allow?)near5(program or executable or software)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
726/2.ccls.and (pre-approv? or approv? or allow?)near5(program or executable or software)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
726/14.ccls.and (pre-approv? or approv? or allow?)near5(program or executable or software)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
726/22.ccls.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
726/24.ccls.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
726/26.ccls.	PGPB, USPT, EPAB, JPAB,	OR	YES		07-18-2008

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1369

	DWPI, TDBD				
713/161.ccls.and (pre-approv? or approv? or allow?)near5(program or executable or software)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
713/164.ccls.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
713/165.ccls.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
713/182.ccls.and (pre-approv? or approv? or allow?)near5(program or executable or software)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
713/187.ccls.and (pre-approv? or approv? or allow?)near5(program or executable or software)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
713/188.ccls.and (pre-approv? or approv? or allow?)near5(program or executable or software)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
713/167.ccls.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
Thomas near James near Satterlee.in.		OR	YES		07-18-2008

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1370

	PGPB, USPT, EPAB, JPAB, DWPI, TDBD				
William near Frank near Hackenberger.in.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
Internet near Security near Systems.as. and (execut?)near4(program)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-18-2008
TPM and sandbox	PGPB	OR			07-18-2008
sandbox	PGPB	OR			07-18-2008
(allow$4)near5(new)with(program or application or executable or software or code)with(run$4 or execut$4) and (sandbox)	PGPB	OR			07-18-2008
(sandbox) and (monitor$4)with(new)with(program or software or code or executable or application)	PGPB	OR			07-18-2008
(sandbox) and Gong.in.	PGPB	OR			07-18-2008
(sandbox) and (allow$4)with(execut$4)near5(computer)	PGPB	OR			07-18-2008
(sandbox) and (applet)	PGPB	OR			07-18-2008
(allowed or approv$4 or authoriz$4)near5(program or software or application or executable or code)near5(run$4 or execut$4)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR			07-18-2008
(new)near5(program or executable or software or application or code)near5(compar$4 or validat$4 or authenticat$4) and ((allowed or approv$4 or authoriz$4)near5(program or software or application or executable or code)near5(run$4 or execut$4))	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR			07-18-2008
((new)near5(program or executable or software or application or code)near5(compar$4 or validat$4 or	PGPB, USPT,	OR			07-18-2008

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1371

authenticat$4) and (allowed or approv$4 or authoriz$4)near5(program or software or application or executable or code)near5(run$4 or execut$4)) and sandbox	EPAB, JPAB, DWPI, TDBD				
(new)near5(program or executable or software or application or code)near5(compar$4 or validat$4 or authenticat$4)with(allow$4 or approv$4)near5(program or software or application or executable or code)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR			07-18-2008
(new or unknown)with(program or software or application or code or applet or activex)near5(non-allow$4 or non-approv$4)with(program or software or application or code or applet or activex)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR			07-18-2008
((allowed or approv$4 or authoriz$4)near5(program or software or application or executable or code)near5(run$4 or execut$4)) and (sandbox)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR			07-18-2008
(new or unknown)near5(no near4 validat$4 or no near4 authoriz$4 or no near5 authenticat$4)with(program or software or application or code) and ((allow$4)near5(new)with(program or application or executable or software or code)with(run$4 or execut$4) and sandbox)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR			07-18-2008
(new or unknown)near5(no near4 validat$4 or no near4 authoriz$4 or no near5 authenticat$4)with(program or software or application or code) and ((allowed or approv$4 or authoriz$4)near5(program or software or application or executable or code)near5(run$4 or execut$4) and sandbox)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR			07-18-2008
(execut$4 or run$4)near5(new)with(program or software or executable or code or application)near5(kernel)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
(monitor$4)near5(new)with(program or software or code application)near5(execut$4 or run$4)with(kernel)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1372

(new)near5(program or executable or software or application or code)near5(compar$4 or validat$4 or authenticat$4)and(allowed or approv$4 or authoriz$4)near5(program or software or application or executable or code)near5(run$4 or execut$4)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
(allow$4 or approv$4 or authoriz$4)near5(program or software or application or code)with(execut$4 or run$4)near5(computer or pc or computing near device or workstation)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
(new or unknown)with(program or software or code or application)near5(execut$4 or run$4)with(computer or pc or computing near device or workstation)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
(compar$4 or validat$4 or authenticat$4 or verif$5)near5(new or unknown)with(program or software or code or application)near5(allow$4 or approv$4 or authoriz$4)with(program or software or application or code) and ((new or unknown)with(program or software or code or application)near5(execut$4 or run$4)with(computer or pc or computing near device or workstation))	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
726/1.ccls. and (pre-approv$3 or approv$3 or allow$3)near5(program or software or code or application)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
726/2.ccls.and (pre-approv$3 or approv$3 or allow$3)near5(program or software or code or application)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
726/22.ccls.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
726/24.ccls.	PGPB, USPT, EPAB,	OR	YES		01-14-2009

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1373

	JPAB, DWPI, TDBD				
726/26.ccls.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
713/161.ccls.and (pre-approv$3 or approv$3 or allow$3)near5(program or code or software or application)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
726/14.ccls.and (pre-approv$3 or approv$3 or allow$3)near5(program or code or software or application)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
713/161.ccls. and (pre-approv$4 or approv$4 or allow$3)near5(program or software or application or code)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
713/164.ccls.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
713/165.ccls.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
713/182.ccls.and (pre-approv$4 or approv$4 or allow$3)near5(program or software or application or code)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1374

713/187.ccls.and (pre-approv$4 or approv$4 or allow$3)near5(program or software or application or code)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
713/188.ccls.and (pre-approv$4 or approv$4 or allow$3)near5(program or software or application or code)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
713/167.ccls.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
Thomas near James near Satterlee.in.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
William near Frank near Hackenberger.in.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
Internet near Security near Systems.as. and (execut$4)near4(program)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		01-14-2009
(new or unknown or untrusted or unapprov$6 or non adj allow$8)near(no near validat$6 or no near authorizat$6 or no near authenticat$5)near(program or software or application or code)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-14-2009
(monitor$5)near(new)with(program or software or code or application)with(kernel)	PGPB, USPT, EPAB, JPAB,	OR	YES		07-14-2009

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1375

	DWPI, TDBD				
(execut$5 or run$5)near(new)with(program or software or executable or code or application)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-14-2009
(execut$5 or run$5)near(new)near(program or software or executable or code or application)near(kernel)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-14-2009
726/22.ccls.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-14-2009
726/24.ccls.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-14-2009
726/26.ccls.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-14-2009
713/164.ccls.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-14-2009
713/165.ccls.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-14-2009
		OR	YES		07-14-2009

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1376

713/182.ccls. and (execut$5 or run$5)near(new or untrust$5 or unknown)with(program or software or executable or code or application)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD				
726/1.ccls. and (execut$5 or run$5)near(new or untrust$5 or unknown)with(program or software or executable or code or application)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-14-2009
726/2.ccls.and (execut$5 or run$5)near(new or untrust$5 or unknown)with(program or software or executable or code or application)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-14-2009
713/161.ccls. and (execut$5 or run$5)near(new or untrust$5 or unknown)with(program or software or executable or code or application)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-14-2009
726/14.ccls.and (execut$5 or run$5)near(new or untrust$5 or unknown)with(program or software or executable or code or application)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-14-2009
713/187.ccls.and (execut$5 or run$5)near(new or untrust$5 or unknown)with(program or software or executable or code or application)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-14-2009
713/188.ccls.and (execut$5 or run$5)near(new or untrust$5 or unknown)with(program or software or executable or code or application)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-14-2009
713/167.ccls.	PGPB, USPT, EPAB, JPAB,	OR	YES		07-14-2009

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1377

	DWPI, TDBD				
Thomas near James near Satterlee.in.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-14-2009
William near Frank near Hackenberger.in.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-14-2009
Internet near Security near Systems.as. and (execut$5)near(new)with(program)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-14-2009
IBM.as. and(execut$5)near(new)with(program)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-14-2009
713/164.ccls. and (determin$5 near new near program near allowed near program and monitor$5 near new near program near execut$6 and execut$5 near kernel).CLM.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-14-2009
713/165.ccls.and (determin$5 near new near program near allowed near program and monitor$5 near new near program near execut$6 and execut$5 near kernel).CLM.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-14-2009
713/167.ccls.and 713/165.ccls.and (determin$5 near new near program near allowed near program and monitor$5 near new near program near execut$6 and execut$5 near kernel).CLM.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-14-2009
		OR	YES		07-14-2009

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1378

726/22.ccls.and (determin$5 near new near program near allowed near program and monitor$5 near new near program near execut$6 and execut$5 near kernel).CLM.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD				
726/24.ccls.and (determin$5 near new near program near allowed near program and monitor$5 near new near program near execut$6 and execut$5 near kernel).CLM.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-14-2009
726/26.ccls.and (determin$5 near new near program near allowed near program and monitor$5 near new near program near execut$6 and execut$5 near kernel).CLM.	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR	YES		07-14-2009
(execut$6 or run$5)near(new)near(program or code or software or executable or application)with(kernel)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR			07-14-2009
(execut$5 or run$5)near(untrusted or unknown or unapprov$5 or non adj allow$8)near(program or code or software or executable or application)with(kernel)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR			07-14-2009
(new or unknown or untrusted or unapprov$6 or non adj allow$8)near(no near validat$6 or no near authorizat$6 or no near authenticat$5)near(program or software or application or code)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR			07-14-2009
(monitor$5)near(new)with(program or software or code or application)with(kernel)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR			07-14-2009
(execut$5 or run$5)near(new)with(program or software or executable or code or application)	PGPB, USPT, EPAB, JPAB,	OR			07-14-2009

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1379

	DWPI, TDBD				
(execut$5 or run$5)near(new)near(program or software or executable or code or application)near(kernel)	PGPB, USPT, EPAB, JPAB, DWPI, TDBD	OR			07-14-2009

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1380

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. 7850

SERIAL NUMBER	FILING or 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
10/336,299	01/03/2003 RULE	713	2439	05456.105038

APPLICANTS

Thomas James Satterlee, Felton, CA;
William Frank Hackenberger, Los Altos, CA;

/JENISE JACKSON/

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

This appln claims benefit of 60/345,432 01/04/2002

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

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

02/27/2003

Foreign Priority claimed	❑ Yes ☑ No	❑ Met after Allowance	STATE OR COUNTRY	SHEETS DRAWINGS	TOTAL CLAIMS	INDEPENDENT CLAIMS
35 USC 119(a-d) conditions met	❑ Yes ☑ No		CA	8	40	5
Verified and Acknowledged	/JENISE E JACKSON/ Examiner's Signature	Initials				

ADDRESS

KING & SPALDING, LLP
INTELLECTUAL PROPERTY DEPT. - PATENTS
1180 PEACHTREE STREET, N.E.
ATLANTA, GA 30309-3521
UNITED STATES

TITLE

System and method for the managed security control of processes on a computer system

FILING FEE RECEIVED		
1408	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

IPR2024-00027
CrowdStrike Exhibit 1002 Page 1381

UNITED STATES DEPARTMENT OF COMMERCE
U.S. Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450

APPLICATION NO./ CONTROL NO.	FILING DATE	FIRST NAMED INVENTOR / PATENT IN REEXAMINATION	ATTORNEY DOCKET NO.
10336299	1/3/03	SATTERLEE ET AL.	05456.105038

KING & SPALDING, LLP
INTELLECTUAL PROPERTY DEPT. - PATENTS
1180 PEACHTREE STREET, N.E.
ATLANTA, GA 30309-3521

EXAMINER
JENISE E. JACKSON

ART UNIT	PAPER
2439	20090930

DATE MAILED:

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

Commissioner for Patents

The Examiner has considered IDS 6/21/03.

/Edan Orgad/
Supervisory Patent Examiner, Art Unit 2439

PTO-90C (Rev.04-03)

OIPE
JAN 21 2003
PATENT & TRADEMARK OFFICE
JC58

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	)
	)
Satterlee et al.	)
	)
Serial No.: **10/336,299**	)
	)
Filed: **January 3, 2003**	)
	)
For: **System and Method for the Managed Security Control of Processes on a Computer System**	)

INFORMATION DISCLOSURE STATEMENT

Commissioner for Patents
Washington, DC 20231

Sir:

The citation of information on the attached Form PTO-1449, "List of Art Cited by Applicant" is made pursuant to 37 C.F.R. §§ 1.56, 1.97, and 1.98. A copy of each cited item is enclosed.

The citation of this information does not constitute an admission of priority or that any cited item is available as a reference, or a waiver of any right the applicant may have under applicable statutes, Rules of Practice in patent cases, or otherwise.

Respectfully submitted,

Robert T. Neufeld
Reg. No. 48,394

King & Spalding LLP
45th Floor, 191 Peachtree Street, N.E.
Atlanta, GA 30303
404.572.4600
K&S Docket: 05456.105038 US
3088899 v1

I certify that this correspondence is being deposited with the United States Postal Service in an envelope addressed to: Commissioner for Patents, Washington, DC 20231, on January 15, 2003.

Robert T. Neufeld, Reg. No. 48,394

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /JJ/