(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
11 March 2004 (11.03.2004)

PCT

(10) International Publication Number
WO 2004/021197 A1

(51) International Patent Classification[7]: G06F 12/14, 11/30

(21) International Application Number: PCT/US2003/026993

(22) International Filing Date: 26 August 2003 (26.08.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/231,557 30 August 2002 (30.08.2002) US
10/647,644 25 August 2003 (25.08.2003) US

(71) Applicant: **WHOLESECURITY, INC.** [US/US]; 5001 Plaza on the Lake, Suite 301, Austin, TX 78746 (US).

(72) Inventors: **OBRECHT, Mark, Eric**; 2301 South Mopac #734, Austin, TX 78746 (US). **ALAGNA, Michael, Anthony**; 4424 Gaines Ranch Loop #130, Austin, TX 78735 (US). **PAYNE, Charles, Andrew**; 7601 Rialto Boulevard #1736, Austin, TX 78735 (US).

(74) Agents: **DAVIS, JR., Michael, A.** et al.; Haynes and Boone, LLP, 901 Main Street, Suite 3100, Dallas, TX 75202-3789 (US).

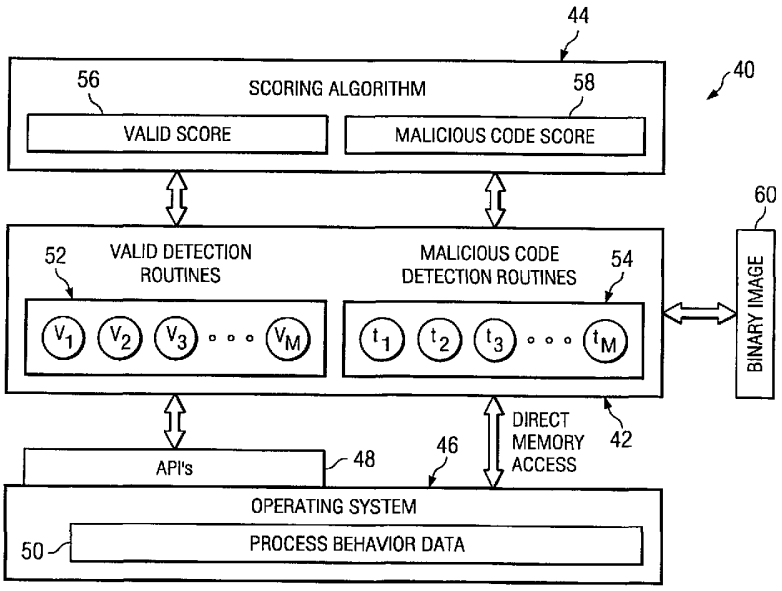
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

[Continued on next page]

(54) Title: METHOD AND APPARATUS FOR DETECTING MALICIOUS CODE IN AN INFORMATION HANDLING SYSTEM



(57) Abstract: Malicious code detection code is executed by an information handling system (10). The malicious code detection code includes detection routines (54). The detection routines (52) are applied to executable code under investigation. The detection routines associate weights to respective code under investigation in response to detections of a valid program or malicious code as a function of the detection routines. It is determined whether code under investigation is a valid program or malicious code as a function of the weights associated by the detection routines.

WO 2004/021197 A1

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

METHOD AND APPARATUS FOR DETECTING MALICIOUS CODE IN AN INFORMATION HANDLING SYSTEM

Background

The present disclosure relates generally to information handling systems, and more particularly to a method and apparatus for detecting malicious code in an information handling system.

Malicious code is computer software code that is executed by an information handling system, typically a computer (but it can also be a Personal Digital Assistant, embedded system, or other information handling device), and can be used for malicious purposes, such as damaging, altering or using the system without permission or knowledge of the system's owner or user, even if the code also has legitimate purposes. There are many different types of malicious code, such as trojan horses, remote control software, keystroke loggers, spyware, worms, viruses, and monitoring software.

Accordingly, a need has arisen for a method and apparatus for detecting malicious code in an information handling system, in which various shortcomings of previous techniques are overcome.

Summary

Malicious code detection code is executed by an information handling system. The malicious code detection code includes detection routines. The detection routines are applied to executable code under investigation. The detection routines associate weights to respective code under investigation in response to detections of a valid program or malicious code as a function of the detection routines. It is determined whether code under investigation is a valid program or malicious code as a function of the weights associated by the detection routines.

It is a technical advantage that various shortcomings of previous techniques are overcome.

Brief Description of the Drawing

Figure 1 is a system block diagram of an information handling system for detecting malicious code, according to one embodiment of the present disclosure; and

Figure 2 is a process diagram of a detection architecture of a malicious code detection program, according to one embodiment of the present disclosure.

Detailed Description

FIG. 1 is a system block diagram of an information handling system 10 (or "computer" or "computer system" or "machine") for detecting malicious code, according to one embodiment of the present disclosure. Although the present disclosure describes some of the most common forms of malicious code, the present disclosure relates to all forms of malicious code.

Malicious code is computer software code that is executed by an information handling system and can be used for malicious purposes, such as damaging, altering or using the system without permission or knowledge of the system's owner or user, even if the code also has legitimate purposes. For example, a remote control program can be used by a system administrator to perform legitimate operations on another user's computer, but the remote control program may nevertheless be considered malicious code, because it can also be used for malicious purposes. Code is embodied in the form of one or more executable instructions and/or their associated operands for an information handling system ("programs" or "computer programs"), according to a variety of techniques, such as an independent program, a library, a thread, a routine or subroutine, or an operating system component, any of which can be written in any computer programming language (e.g., scripting languages, interpreted languages, compiled languages, assembly languages or machine code).

Malicious code is stored in any computer-readable medium, such as a hard disk drive, floppy diskette, CD-ROM, DVD or memory. During operation of an information handling system, malicious code has one or more states, such as active, inactive, executing (or "running"), not executing, hidden or visible. In the illustrative embodiments, the malicious code detection program is operable to detect malicious code, irrespective of the malicious code's states, and irrespective of the computer-readable media storing the malicious code.

Trojan horses ("trojans") are a particular type of malicious code. The trojan is executable code that exists in a variety of different forms. For example, some (but not all) forms of trojans are instantiated in executable code as one or more programs, threads inside other programs, plugins or shared modules loaded by other programs, or modules loaded into operating system kernel memory in the manner of a device driver or loadable kernel module. A trojan is a form of malicious code that enables a person to remotely control someone else's computer. The person who remotely controls the computer is known as the "Evil Hacker" (or "hacker") while the person whose computer is being remotely controlled is known as the "Innocent Victim" (or "victim"). BackOrifice2000, SubSeven, NetBus and OptixPro are all examples of trojans.

Trojans are sometimes referred to as "back-doors" or "hacker back-doors."

Most trojans have two components, the client program (trojan client) that is executed by the evil hacker's computer and the server program (trojan server) that is executed by the innocent victim's computer. Some trojans have only a trojan server that can be remotely controlled through manually entered commands rather than through the programmatic interface of a trojan client.

There are many ways to infect a computer with a trojan including sending the innocent victim the trojan server disguised as a valid program, copying the trojan server onto the innocent victim's computer, or exploiting a vulnerability in the innocent victim's computer to place the trojan server on the computer.

Several techniques exist that are effective for detecting some forms of malicious code. For example, some types of malicious code can be detected by examining the binary code image of the program during its execution or the binary code image of the program when it is stored on a storage device. Many malicious code programs can be identified by a unique bit or byte pattern. The unique bit or byte pattern can include the entire image of the program while it is stored in memory or while it is stored on disk. The signature can also be a bit or byte pattern that is a portion of the program in memory or on disk. Once the unique sequence has been identified, a signature can be developed to identify the sequence. The signature is often the bit or byte pattern itself or it is in the form of a checksum. A detection program can then search for a malicious code program using the signature to identify the unique bit or byte sequence. Trojans, however, may be configurable to have no easily identifiable signature. Trojans may have configuration parameters that change the bit or byte sequences in the program and make it difficult or impossible to provide a unique signature. Various tools can be used to reconfigure a trojan, so that it will not have a known signature.

Another technique used to identify trojans examines the behavior of a trojan server while the trojan server is loaded and installed on a computer. With such a technique, a loaded and installed program is first placed into a sandbox, which includes a restricted area on the computer where the program (e.g., trojan server) can be examined safely. While such an approach may be effective for preventing some trojan infection, the approach does not however detect trojan servers once they are already installed on a computer. Such an approach does not detect many trojan servers because trojans do not exhibit their most characteristic behaviors while they are

being loaded or installed, but rather they come alive and exhibit their malicious behavior after they have been loaded and installed.

Remote control software ("remote control"), such as pcAnywhere and VNC, is another type of malicious code, which has much of the same functionality as trojans. These programs allow for remote administration (via a "client" on a host personal computer ("PC")) of a target PC that is executing the "server" portion of the program. A goal of a trojan is to be stealth and transparent to the innocent victim, so as to remotely control the PC or other machine. By comparison, a goal of remote controls is to allow a trusted remote user to administer the machine for efficiency purposes. Nevertheless, remote controls can also be used by an evil hacker to remotely control a machine that is "infected" by the unauthorized remote control, in a stealthy and malicious manner. Moreover, even if a remote control is operated by a trusted legitimate user, the remote control can also be used by malicious individuals if proper security precautions are not taken (e.g., password protection, authentication, encryption). Accordingly, remote controls can be used for malicious purposes, so the present disclosure relates to them as well.

Keystroke loggers ("keyloggers" or alternatively "keyboard loggers") are another type of malicious code. The keylogger is executable code that can exist in one of many forms. For example, some forms of keyloggers can be instantiated in executable code as one or more programs, computer files, threads inside other programs, plugins or shared modules loaded by other programs, or modules loaded into operating system kernel memory in the manner of a device driver or loadable kernel module. A keylogger is a form of malicious code that enables a person to obtain the actual "punched" keystrokes from an infected computer. A record of the keystrokes usually exists in the form of a file on the file system, which stores the "punch for punch" results of what was typed at the keyboard. Also, some keyloggers provide capability for e-mailing (to an e-mail address) a record of the captured keystrokes, in order to share access and remotely identify the typed characters. Alternate access mediums are sometimes used for obtaining a record of the keystrokes, such as physical access to the infected system, e-mailing a file to a configured e-mail account, or "backdoor" access to the machine via a trojan. Sinred, Fearless KeySpy, and TeeJayEm KeySpy are examples of keyloggers. Typically, a keylogger is a software application (e.g., which may be, but is not necessarily, a standalone application) that exists in a machine's operating system.

Monitoring software is another type of malicious code, which has many similarities to keyloggers. In many respects, monitoring sof e performs operations that are similar to a

keylogger. Monitoring software is often used to monitor a wide range of the computer's activity. For example, monitoring software is useful for a person to obtain a recorded log of actions that children, a spouse, friends, co-workers and others perform on their computers. Unlike keyloggers, monitoring software is often, but not always, purchased from a software vendor and installed by the computer's owner to achieve a new layer of surveillance over the computer owner's machine.

Spyware is an Internet term for advertising supported software ("adware"). Spyware differs from other malicious code, because spyware has legitimate purposes, in addition to potentially malicious purposes. Spyware is installed in a computer, according to a variety of techniques. Spyware's primary purpose is the gathering of marketing and statistical information about a user's electronic behavior, together with the reporting of such information via the infected machine's Internet connection to one or more collection servers via the Internet. According to the privacy policies of many advertising companies that develop and distribute spyware, no sensitive information (or other information that identifies the individual user) is authorized to be collected from the user's computer. Such a policy is helpful to allay possible concerns regarding invasion of privacy or malicious purpose. Nevertheless, such policies are not always adopted or followed. Many spyware examples contain a "live" server program executed by the machine, which is capable of sending personal information and web-surfing habits to a remote location. Accordingly, spyware is also covered by the present disclosure, because spyware can be used for malicious purposes.

Spyware has resulted in congestion of Internet web pages, as an increasingly large number of vendors create and distribute spyware via Internet sites that attract visitors. Spyware has also become a popular technique for shareware authors to profit from a product, other than by selling it directly to users. For example, if a user prefers, it can freely install an application bundled with spyware, instead of purchasing a license to the application. Several large media companies offer to place banner advertisements in software products, in exchange for a portion of revenue from sales resulting from the software products' display of the banner. This technique has increased in popularity, because users can avoid paying for the software products, and the software product developers receive money from alternate sources. If the user is annoyed by the banners, the user is usually given an option to remove them by paying a regular licensing fee for the software products.

Spyware is not illegal, but it raises various privacy issues for certain users. Such privacy

issues are raised when the spyware tracks and sends information and statistics via a private Internet connection that operates in the "background" of the user's PC, using a server program that is installed on the user's PC. In a written privacy statement, legitimate adware companies will disclose the nature of such information that is collected and transmitted, but the user is typically unable to actually control it.

Worms are another type of malicious code that exists in a variety of different forms. For example, some (but not all) forms of worms are instantiated in executable code as one or more programs, computer files, threads inside other programs, plugins or shared modules loaded by other programs, or modules loaded into operating system kernel memory in the manner of a device driver or loadable kernel module. Worms are distributed ("spread") via a computer network, such as the Internet. From the computer network, they penetrate a computer's memory, calculate network addresses of other computers, and send copies of themselves to such addresses for additional replication. Worms often exploit OS, application or service vulnerabilities to propagate themselves and penetrate remote machines. Worms have various purposes, designs, propagation media, and techniques for exploiting vulnerabilities. On the machine, worms may deposit a "payload," which performs some or no operation. Frequently, this payload includes a trojan or keylogger. Examples of worms are Code Red and Sircam. Worms are convenient vehicles for evil hackers to distribute other types of malicious code.

Viruses are another type of malicious code that can exist in a variety of different forms, such as macro viruses, boot sector viruses, and parasitic viruses. For example, some (but not all) forms of viruses are instantiated in executable code are as one or more programs, computer files, threads inside other programs, plugins or shared modules loaded by other programs, or modules loaded into operating system kernel memory in the manner of a device driver or loadable kernel module. Some viruses merely replicate by inserting (or attaching) themselves to a medium, in order to infect another program, boot sector, partition sector, or document that supports macros. But many viruses additionally inflict a large amount of damage on the machine. On the machine, viruses may deposit a payload, which performs some or no operation. Frequently, this payload includes a trojan or keylogger.

Malicious code, such as trojans, keyloggers, worms and viruses, can be used by evil hackers to disrupt the normal operation of the innocent victim's computer, to spy on the innocent victim, to steal money from the innocent victim, or to steal intellectual property from the innocent victim. The evil hacker often uses tl · nocent victim's computer to perform these

malicious activities, in order to harm the innocent victim's associated organization (e.g., company or government). Accordingly, such malicious code can harm a computer system, irrespective of whether the computer system belongs to an individual or an organization.

Information handling system 10 includes one or more of: a central processing unit (CPU) 12, memory 14, input/output (I/O) devices, such as a display, a keyboard, a mouse, and associated controllers, collectively designated by a reference numeral 16, a hard disk drive 18, or other storage devices or media drives, such as a floppy disk drive, a CD-ROM drive, a DVD drive, and memory device, collectively designated by a reference numeral 20, and/or various other subsystems, such as a network interface card or wireless communication link (collectively designated by a reference numeral 22), all interconnected, for example, via one or more buses (shown collectively as a bus 24). Examples of information handling systems are a personal computer system, a personal digital assistant, a thin client device, a thick client device, or similar information handling device.

In one embodiment, the information handling system (IHS) 10 is configured with a suitable operating system for installing and executing instructions from one or more computer readable media 26, such as a hard disk drive, floppy diskette, CD-ROM, DVD, or memory. Information handling system 10 may further be configured for communicating with another information handling system 28 (e.g., through a network 30 via a suitable communication link or links). The operating system of IHS 10 may optionally include instructions for installing and executing programs, and for downloading information via network 30. The illustrative embodiments of the present disclosure may be practiced over an intranet, the Internet, virtual private network, or other suitable communication network.

According to one embodiment, the technique for malicious code detection is implemented in the form of computer software, the computer software including instructions executable by the CPU of a computer system, such as an innocent victim's computer system. The instructions include suitable program code processable by the computer system for performing the various functions as described herein. The various functions as discussed herein can be programmed using programming techniques well known in the art.

One technique for detecting malicious code includes a technique for detecting the portion of the malicious code that resides on a target computer system, such as an innocent victim computer system. For some forms of malicious code, such as keyloggers and Viruses, all of the malicious code resides on the innocent victim ' mputer system. For other forms of malicious

code, such as trojans and remote controls, only the server portion of the malicious code resides on the innocent victim's computer system. The procedure can be embodied in a computer program, such as a malicious code detection program. The malicious code detection program detects the presence of (and identifies) the malicious code before, during and/or after the malicious code executes on the victim's computer system.

Figure 2 illustrates an architecture of a malicious code detection program 40 that is executed by the information handling system 10 according to an embodiment of the present disclosure. The malicious code detection program 40 includes detection routines 42 and a scoring algorithm 44. The detection routines 42 operatively couple to the operating system 46 of the computer system under investigation via application programming interfaces (APIs) 48. The detection routines also access process behavior information (e.g., data) 50 and binary image information 60, according to the particular requirements of a corresponding detection routine, further as discussed below.

In one embodiment, the malicious code detection program operates as follows. The malicious code detection program executes at any time, on an as-needed basis, a periodic basis, a random basis, another scheduled basis, or on an event driven basis in response to a particular event according to the particular requirements of a given situation. In the illustrative embodiments, the malicious code detection program includes instructions for the information handling system to examine characteristics and behaviors of the information handling system's instructions and/or data.

The malicious code detection program includes instructions for the information handling system to evaluate the information handling system's instructions and/or data to determine whether such instructions and/or data are valid code (i.e., non-malicious) or malicious code or any one or more types. The malicious code detection program includes respective detection routines, sets of weights, and weighted scoring algorithms for detecting one or more types of valid code and/or one or more types of malicious code.

The malicious code detection program 40 contains detection routines 42, including valid program detection routines 52 and malicious code detection routines 54. The valid program detection routines 52 include one or more routines identified by v_1, v_2, v_3, ... , v_M in Figure 2. The valid program detection routines 52 are configured to determine whether the program under investigation has characteristics and behaviors usually associated with a valid program. The malicious code detection routines 54 include one or more routines identified by t_1, t_2, t_3, ... , t_N in

Figure 2. The malicious code detection routines 54 are configured to determine whether the program under investigation has characteristics and behaviors usually associated with a malicious code program.

In one embodiment, the valid program detection routines 52 and the malicious code detection routines 54 are configured to gather a variety of characteristic and behavior information from the information handling system in a variety of ways, such as: (a) examining the program itself; (b) accessing information from the operating system 46 using application programming interfaces (APIs) 48 to the operating system (including documented APIs and/or undocumented API's); (c) kernel and/or device driver interfacing; and/or (d) direct access to resources of the information handling system such as memory, network connections, storage media, and/or other devices. For example, as shown in Fig. 2, the detection routines 42 gather such information by examining one or more of (a) a binary image 60 or (b) a library or other information (e.g., tables showing a program's network connection activity) that indicates the aforementioned characteristics and behaviors, such as process behavior information 50.

For example, a detection routine 42 can be configured to account for the following. Many trojans, keyloggers, remote controls and monitoring software programs log keystrokes on the innocent victim's computer and transmit the keystroke information from the innocent victim's computer to the evil hacker's computer. In one embodiment, a malicious code detection routine 54 determines whether the program being examined is logging keystrokes. Since there are many different ways for a program to log keystrokes, one or more of the malicious code detection routines 54 can be configured to examine the program under investigation to determine whether the program is using any of a number of different mechanisms for logging keystrokes. Detection routines may output many different types of results, such as numeric values, boolean values, counts or lists.

The malicious code detection program 40 further includes a scoring algorithm 44. In the illustrative embodiment, the scoring algorithm calculates two scores, namely a valid program score 56 and a malicious code score 58. In an alternative embodiment, the scoring algorithm calculates the valid program score 56, but not the malicious code score 58. In another alternative embodiment, the scoring algorithm calculates the malicious code score 58, but not the valid program score 56.

If the result of a valid program detection routine 52 indicates that the characteristic or behavior of the program being examined was [illegible] of a valid program, then a weight, W_i, is

associated with the routine and that weight contributes positively to the valid program score 56. A weight, W_i, is assigned to each valid program detection routine, for i = 1 to M, where M is the number of the valid program detection routine.

The weight indicates (a) the detection routine's importance, (b) the extent to which the particular behavioral trait being measured by the detection routine is present, and (c) the extent to which the behavioral trait contributes to the determination of whether the program is valid or malicious. To determine the value that results from combining the weight with the results of the detection routine, the information handling system performs any one or more of a variety of operations, such as performing an arithmetic or algebraic operation on the combination of the weight and the result of the detection routine or simply assigning the combination a numerical value.

If the result of a malicious code detection routine 54 indicates that the characteristic or behavior of the program being examined was that of a malicious code program, then a weight, W_j, is associated with the routine and that weight contributes positively to the malicious code score 58. A weight, W_j, is assigned each malicious code detection routine, for j = 1 to N, where N is the number of the malicious code detection routine.

According to one embodiment, the scoring algorithm 44 includes an algorithm that has an algebraic formula for determining the two scores 56 and 58. The scoring algorithm is dependent on the valid program detection routines 52 and the weights, W_i, associated with each valid program detection routine, in addition to, the malicious code detection routines 54 and the weights W_j, associated with each malicious code detection routine. The algebraic formula or equation can also be made arbitrarily complex (e.g., associating additional weights to one or more to combinations of detection routines 42.

In one embodiment, the scoring algorithm 44 includes an algebraic equation defined as a sum of weighted values. For example, the algebraic equation for the valid program detection routines can include an equation as given by:

$$VALID\ SCORE = \sum_{i=1}^{M} W_i,$$

where W_i = weight of a valid detection routine v_i for i = 1 to M.

Similarly, the algebraic equation for the malicious code detection routines can include an equation as given by:

$$MALICIOUS\ CODE\ SCORE = \sum_{j=1}^{N} W_j,$$

where W_j = weight of a malicious code detection routine t_j for j = 1 to N.

In another embodiment, more complex forms of the scoring algorithm 44 can be implemented in the form of more sophisticated algebraic formulae.

If a program under investigation exceeds a valid program score threshold, V_{thres}, then it is determined that the program is a valid program. If that program exceeds a malicious code score threshold, T_{thres}, then it is determined that the program is a malicious code program. If a program is deemed to be valid using the valid algorithm, then it is sometimes eliminated from consideration as a malicious code program.

Executable code and/or programs under investigation may also have some of the characteristics and behaviors of valid programs and some of the characteristics and behaviors of malicious code. If a program does not exceed either threshold or if a program does not have a significant difference between the valid program score 56 and the malicious code score 58, then according to another embodiment of the present disclosure, the technique identifies the program in another category of suspicious programs or anomalous programs.

In one embodiment, the technique for detecting malicious code on a computer system includes executing a malicious code detection program on the computer system. The malicious code detection program includes detection routines. The malicious code detection program applies the detection routines to programs on the computer system. The detection routines are assigned weights that are factored by a scoring algorithm to determine a composite score based on the results of the detection routines and their associated weights. For example, a malicious code detection routine has a weight associated with it, such that if the malicious code detection routine determines that a given code under investigation is a malicious code program, then the weight is applied positively towards the malicious code score for the code under investigation. Also, the malicious code detection program determines whether one or more programs are valid or malicious as a function of the weights assigned to the detection routines.

In another embodiment, the technique is configured to detect malicious code on a computer having an operating system. The technique includes executing a malicious code detection program on the computer. Detection routines of the malicious code detection program are configured to gather information about programs on the computer system. The detection routines include at least one selected from the group consisting of (a) examining each executable

code or program itself and (b)searching for information about each executable code or program in the operating system. For example, examining code or a program can include examining a binary image of the same, wherever the binary image may reside, within the IHS or in computer readable media accessible to the IHS. In addition, the detection routines further consist of valid program detection routines and malicious code detection routines.

The malicious code detection program applies the detection routines to the programs on the computer system. In response to a detection of a valid program or malicious code, the detection routines assigns weights to respective programs under test as a function of a respective detection routine. Also, the malicious code detection program determines whether a program is a valid program or malicious code as a function of the weights assigned by the detection routines. Determining whether the program is a valid program or malicious code involves the scoring of an execution of each detection routine as a function of a respective weight. A scoring algorithm is used to identify a program as malicious code in response to a valid score and a malicious code score, as discussed herein.

In yet another embodiment, the technique for detecting malicious code on a computer system includes executing detection routines, the detection routines having been configured to examine at least one selected from the group consisting of characteristics and behaviors of programs on the computer system. For example, the detection routines can be configured to access process behavior information of a program on the computer system. In addition, the characteristics and behaviors may include one or more of logging keystrokes, saving a display screen view, uploading files, downloading files, executing programs, and controlling a display screen of the computer system.

Subsequent to execution of one or more of the detection routine, weights are assigned as a function of the examined characteristics and behaviors, the assigned weights indicative of a valid program or malicious code as a function of respective detection routines. Also, the technique determines whether a program is malicious code as a function of the weights assigned by the detection routines.

In the embodiment of the previous paragraph, the detection routines include valid program detection routines and malicious code detection routines. The valid program detection routines are configured to determine whether the program exhibits at least one or more characteristics and behaviors associated with a valid program. The malicious code detection routines are configured to determine whether ' rogram exhibits at least one or more

characteristics and behaviors associated with malicious code.

In one embodiment, the technique for detecting malicious code is implemented in the form of a computer program. The computer program is executed on a desired computer system for detecting any potential malicious code on the computer system. Execution of the computer program continues until all active programs on the computer system have been tested and evaluated. Alternatively, other criteria may be established for a duration of testing with the malicious code detection program. For example, execution of the malicious code detection program can be configured to occur in response to one or more of a random initiation and a periodic initiation.

According to another embodiment, the malicious code detection program includes a small program configured for being delivered quickly, as well as, for being executed quickly. The malicious code detection program can be delivered to the innocent victim's computer over a network, such as a Local Area Network (LAN), Wide Area Network (WAN), Internet, intranet, or any other global computer network 30. The malicious code detection program may also be delivered via suitable computer readable media, such as, media 26 shown in Figure 1.

While not stopping an infection of the computer system with malicious code programs, the technique of the present embodiments identifies a malicious code program when executing on a computer system. The technique for identifying a malicious code program is suitable for combination with other techniques, such as a technique for detecting infection, resulting in a more robust computer system malicious code protection implementation.

Where the foregoing disclosure mentions that code performs an operation, it is understood that the information handling system performs the operation in response to the information handling system's execution of the code.

Although illustrative embodiments have been shown and described, a wide range of modification, change and substitution is contemplated in the foregoing disclosure and, in some instances, some features of the embodiments may be employed without a corresponding use of other features. Accordingly, all such modifications are intended to be included within the scope of the embodiments. Accordingly, it is appropriate that the appended claims be construed broadly. In the claims, means-plus-function clauses are intended to cover the structures described herein as performing the recited function and not only structural equivalents, but also equivalent structures.

Claims

What is claimed is:

1. A method for detecting malicious code in an information handling system, comprising:

executing malicious code detection code (MCDC) on the information handling system, the MCDC including detection routines;

applying the detection routines to executable code under investigation, the detection routines associating weights to respective code under investigation in response to detections of a valid program or malicious code as a function of the detection routines; and

determining whether code under investigation is a valid program or malicious code as a function of the weights associated by the detection routines.

2. The method of claim 1, wherein the detection routines include valid program detection routines and malicious code detection routines.

3. The method of claim 1, wherein the applying comprises:

applying the detection routines to gather information about the executable code under investigation by at least one of the following: examining the code or program; and searching for information in the information handling system about the code or program.

4. The method of claim 1, wherein determining whether the code under investigation is a valid program or malicious code includes scoring the execution of the detection routines as a function of the weights.

5. The method of claim 4, wherein scoring includes configuring a scoring algorithm to identify code under investigation as malicious code in response to at least one of a valid score and a malicious code score.

6. The method of claim 1, wherein the malicious code includes a trojan horse.

7. The method of claim 1, wherein the malicious code includes remote control software.

8. The method of claim 1, wherein the malicious code includes a keystroke logger.

9. The method of claim 1, wherein the malicious code includes spyware.

10. The method of claim 1, wherein the malicious code includes a worm.

11. The method of claim 1, wherein the malicious code includes a virus.

12. The method of claim 1, wherein the malicious code includes monitoring software.

13. A method for detecting malicio ›de in an information handling system,

comprising:

executing malicious code detection code (MCDC) on the information handling system, the MCDC including detection routines for gathering information about executable code under investigation, the detection routines including at least one of the following: (a) examining the code or program and (b) searching for information in the information handling system about the code or program, the detection routines including valid program detection routines and malicious code detection routines;

applying the detection routines to the executable code under investigation, the detection routines associating weights to respective code under investigation in response to detections of a valid program or malicious code as a function of at least one of the detection routines; and

determining whether code under investigation is a valid program or malicious code as a function of the weights associated by the detection routines, wherein determining whether the code under investigation is a valid program or malicious code includes scoring an execution of the detection routines as a function of the weights, and wherein scoring includes configuring a scoring algorithm to identify code under investigation as malicious code in response to at least one of a valid score and a malicious code score.

14. The method of claim 13, wherein the malicious code includes a trojan horse.

15. The method of claim 13, wherein the malicious code includes remote control software.

16. The method of claim 13, wherein the malicious code includes a keystroke logger.

17. The method of claim 13, wherein the malicious code includes spyware.

18. The method of claim 13, wherein the malicious code includes a worm.

19. The method of claim 13, wherein the malicious code includes a virus.

20. The method of claim 13, wherein the malicious code includes monitoring software.

21. A method for detecting malicious code on a information handling system, comprising:

executing detection routines, the detection routines examining at least one of the following: characteristics and behaviors of executable code under investigation;

assigning weights as a function of the examined characteristics and behaviors, the assigned weights indicative of a valid program or malicious code as a function of the detection routines; and

determining whether executable code under investigation is malicious code as a function of the weights assigned by the detection routines.

22. The method of claim 21, wherein the detection routines include valid program detection routines and malicious code detection routines.

23. The method of claim 21, wherein the valid program detection routines determine whether the executable code under investigation exhibits at least one or more characteristics and behaviors associated with a valid program; and

wherein the malicious code detection routines determine whether the executable code under investigation exhibits at least one or more characteristics and behaviors associated with malicious code.

24. The method of claim 21, wherein determining whether the executable code under investigation is malicious code includes scoring the execution of the detection routines as a function of the weights.

25. The method of claim 24, wherein scoring includes using a scoring algorithm for identifying executable code as malicious code in response to at least one of a valid score and a malicious code score.

26. The method of claim 25, wherein the scoring algorithm determines a valid program by a summation of weights of the valid program detection routines being greater than a valid program weight threshold, and a malicious code by a summation of weights of the malicious code detection routine having a summed value greater than a malicious code weight threshold.

27. The method of claim 26, wherein the scoring algorithm determines an anomalous program by the summation of weights of the valid program detection routines and the summation of weights of the malicious code detection routines both having sums greater than respective thresholds, or less than the respective thresholds.

28. The method of claim 21, and comprising:

operatively coupling the detection routines to an operating system of the information handling system via application programming interfaces (APIs).

29. The method of claim 21, wherein the detection routines access process behavior information of executable code under investigation.

30. The method of claim 21, wherein the characteristics and behaviors include at least one of the following: logging keystrokes, saving a display screen view, uploading files, downloading files, executing programs, and controlling the display screen.

31. The method of claim 21, wherein the detection routines access information about the executable code under investigation from an operating system of the information handling system via Application Programming Interfaces (APIs), and the detection routines gather information from executable code or a program by examining a binary image of the executable code or program, the characteristics and behavior of the executable code or program, and any other related code or programs used by the executable code under investigation.

32. The method of claim 21, and comprising:

delivering malicious code detection code (MCDC) containing the detection routines to the information handling system in a small compact code module via at least one of the following: a computer network, Internet, intranet, extranet, modem line, and prepackaged computer readable storage media.

33. The method of claim 21, wherein execution of the MCDC occurs in response to at least one of the following: a random initiation, an event driven initiation, and a periodic initiation.

34. The method of claim 21, wherein the malicious code includes a trojan horse.

35. The method of claim 21, wherein the malicious code includes remote control software.

36. The method of claim 21, wherein the malicious code includes a keystroke logger.

37. The method of claim 21, wherein the malicious code includes spyware.

38. The method of claim 21, wherein the malicious code includes a worm.

39. The method of claim 21, wherein the malicious code includes a virus.

40. The method of claim 21, wherein the malicious code includes monitoring software.

41. A computer program stored on computer-readable media for detecting malicious code in an information handling system, the computer program including instructions processable by the information handing system for causing the information handling system to:

execute malicious code detection code (MCDC) on the information handling system, the MCDC including detection routines for gathering information about executable code under investigation, the detection routines including at least one of the following: (a) examining the

executable code or program; and (b) searching for information in the information handling system about the executable code or program, the detection routines including at least one of valid program detection routines and malicious code detection routines;

apply the detection routines to the executable code under investigation, the detection routines associating weights to respective code under investigation in response to detections of a valid program or malicious code as a function of at least one of the detection routines; and

determine whether code under investigation is a valid program or malicious code as a function of the weights associated by the detection routines, wherein determining whether the code under investigation is a valid program or malicious code includes scoring an execution of the detection routines as a function of the weights, wherein scoring includes configuring a scoring algorithm to identify code under investigation as malicious code in response to at least one of a valid score and a malicious code score.

42. The computer program of claim 41, wherein the malicious code includes a trojan horse.

43. The computer program of claim 41, wherein the malicious code includes remote control software.

44. The computer program of claim 41, wherein the malicious code includes a keystroke logger.

45. The computer program of claim 41, wherein the malicious code includes spyware.

46. The computer program of claim 41, wherein the malicious code includes a worm.

47. The computer program of claim 41, wherein the malicious code includes a virus.

48. The computer program of claim 41, wherein the malicious code includes monitoring software.

49. A computer program stored on computer-readable media for detecting malicious code in an information handling system, the computer program including instructions processable by the information handling system for causing the information handling system to:

execute detection routines, the detection routines examining at least one of the following: characteristics and behaviors of executable code under investigation;

assign weights as a function of the examined characteristics and behaviors, the assigned weights indicative of a valid program or malicious code as a function of the detection routines; and

determine whether executable code under investigation is malicious code as a function of

the assigned weights.

50. The computer program of claim 49, wherein the detection routines include valid program detection routines and malicious code detection routines.

51. The computer program of claim 49, wherein the valid program detection routines determine whether the executable code under investigation exhibits at least one or more characteristics and behaviors associated with a valid program; and

wherein the malicious code detection routines determine whether the executable code under investigation exhibits at least one or more characteristics and behaviors associated with malicious code.

52. The computer program of claim 49, wherein determining whether the executable code under investigation is malicious code includes scoring the execution of the detection routines as a function of the weights.

53. The computer program of claim 52, wherein scoring includes using a scoring algorithm for identifying executable code as malicious code in response to at least one of a valid score and a malicious code score.

54. The computer program of claim 53, wherein the scoring algorithm determines a valid program by a summation of weights of the valid program detection routines being greater than a valid program weight threshold, and a malicious code by a summation of weights of the malicious code detection routine having a summed value greater than a malicious code weight threshold.

55. The computer program of claim 54, wherein the scoring algorithm determines an anomalous executable code under investigation by the summation of weights of the valid program detection routines and the summation of weights of the malicious code detection routines both having sums greater than respective thresholds, or less than the respective thresholds.

56. The computer program of claim 49, and comprising instructions processable by the information handling system for causing the information handling system to:

operatively couple the detection routines to an operating system of the information handling system via application programming interfaces (APIs).

57. The computer program of claim 49, wherein the detection routines access process behavior information of executable code under investigation.

58. The computer program of claim 49, wherein the characteristics and behaviors include at least one of the following: logging keystrokes, saving a display screen view, uploading files, downloading files, executing programs, and controlling the display screen.

59. The computer program of claim 49, wherein the detection routines access information about the executable code under investigation from an operating system of the information handling system via Application Programming Interfaces (APIs), and the detection routines gather information from executable code or a program by examining a binary image of the executable code or program, the characteristics and behavior of the executable code or program, and any other related code or programs used by the executable code under investigation.

60. The computer program of claim 49, comprising instructions processable by the information handling system for causing the information handling system to:

deliver malicious code detection code (MCDC) containing detection routines to the information handling system in a small compact code module via at least one of the following: a computer network, Internet, intranet, extranet, modem line, and prepackaged computer readable storage media.

61. The computer program of claim 49, wherein execution of the MCDC occurs in response to at least one of the following: a random initiation, an event driven initiation, and a periodic initiation.

62. The computer program of claim 49, wherein the malicious code includes a trojan horse.

63. The computer program of claim 49, wherein the malicious code includes remote control software.

64. The computer program of claim 49, wherein the malicious code includes a keystroke logger.

65. The computer program of claim 49, wherein the malicious code includes spyware.

66. The computer program of claim 49, wherein the malicious code includes a worm.

67. The computer program of claim 49, wherein the malicious code includes a virus.

68. The computer program of claim 49, wherein the malicious code includes monitoring software.

69. An information handling system, comprising:

a memory;

a processor; and

computer-readable code stored by the memory and processable by the processor for detecting malicious code, the computer-readable code including instructions for causing the processor to:

execute malicious code detection code (MCDC) on the information handling system, the MCDC including detection routines for gathering information about executable code under investigation, the detection routines including at least one of the following: (a) examining the executable code or program and (b) searching for information about the executable code or program in the information handling system, the detection routines including valid program detection routines and malicious code detection routines;

apply the detection routines to the executable code under investigation, the detection routines assigning weights to respective executable code under investigation in response to detections of a valid program or malicious code as a function of at least one of the detection routines; and

determine whether executable code under investigation is a valid program or malicious code as a function of the weights associated by the detection routines, wherein determining whether the code under investigation is a valid program or malicious code includes scoring an execution of the detection routines as a function of the weights, and wherein scoring includes configuring a scoring algorithm to identify executable code under investigation as malicious code in response to at least one of a valid score and a malicious code score.

70. The information handling system of claim 69, wherein the malicious code includes a trojan horse.

71. The information handling system of claim 69, wherein the malicious code includes remote control software.

72. The information handling system of claim 69, wherein the malicious code includes a keystroke logger.

73. The information handling system of claim 69, wherein the malicious code includes spyware.

74. The information handling system of claim 69, wherein the malicious code includes a worm.

75. The information handling system of claim 69, wherein the malicious code includes a virus.

76. The information handling system of claim 69, wherein the malicious code includes monitoring software.

77. An information handling system, comprising:

a memory;

a processor; and

computer-readable code stored by the memory and processable by the processor for detecting malicious code on the information handling system, the computer-readable code including instructions for causing the processor to:

execute detection routines, the detection routines examining at least one of the following:

characteristics and behaviors of programs;

assign weights as a function of the examined characteristics and behaviors, the assigned weights indicative of a valid program or malicious code as a function of the detection routines; and

determine whether executable code under investigation is malicious code as a function of the weights assigned by the detection routines.

78. The information handling system of claim 77, wherein the detection routines include valid program detection routines and malicious code detection routines.

79. The information handling system of claim 77, wherein the valid program detection routines determine whether the executable code under investigation exhibits at least one or more characteristics and behaviors associated with a valid program; and

wherein the malicious code detection routines determine whether the executable code under investigation exhibits at least one or more characteristics and behaviors associated with malicious code.

80. The information handling system of claim 77, wherein determining whether the executable code under investigation is malicious code includes scoring the execution of the detection routines as a function of the weights.

81. The information handling system of claim 80, wherein scoring includes using a scoring algorithm for identifying executable cc ' s malicious code in response to a valid score

and a malicious code score.

82. The information handling system of claim 81, wherein the scoring algorithm determines a valid program by a summation of weights of the valid program detection routines being greater than a valid program weight threshold, and a malicious code by a summation of weights of the malicious code detection routine having a summed value greater than a malicious code weight threshold.

83. The information handling system of claim 82, wherein the scoring algorithm determines an anomalous executable code under investigation by the summation of weights of the valid program detection routines and the summation of weights of the malicious code detection routines both having sums greater than respective thresholds, or less than the respective thresholds.

84. The information handling system of claim 77, wherein the characteristics and behaviors include at least one of the following: logging keystrokes, saving a display screen view, uploading files, downloading files, executing programs, and controlling the display screen.

85. The information handling system of claim 77, wherein the detection routines access information about the executable code under investigation from an operating system of the information handling system via Application Programming Interfaces (APIs), and the detection routines gather information from executable code or a program by examining a binary image of the executable code or program, the characteristics and behavior of the executable code or program, and any other related code or programs used by the executable code under investigation.

86. The information handling system of claim 77, wherein the computer-readable code includes instructions for delivering the MCDC containing detection routines to the information handling system in a small compact code module via at least one of the following: a computer network, Internet, intranet, extranet, modem line, and prepackaged computer readable storage media.

87. The information handling system of claim 77, wherein execution of the MCDC occurs in response to at least one of the following: a random initiation, an event driven initiation, and a periodic initiation.

88. The information handling system of claim 77, wherein the malicious code includes a trojan horse.

89. The information handling syste claim 77, wherein the malicious code

includes remote control software.

90. The information handling system of claim 77, wherein the malicious code includes a keystroke logger.

91. The information handling system of claim 77, wherein the malicious code includes spyware.

92. The information handling system of claim 77, wherein the malicious code includes a worm.

93. The information handling system of claim 77, wherein the malicious code includes a virus.

94. The information handling system of claim 77, wherein the malicious code includes monitoring software.

95. A method for detecting malicious code in an information handling system, comprising:

executing malicious code detection code (MCDC) on the information handling system, the MCDC including detection routines;

applying the detection routines to code under investigation, the detection routines associating weights to respective code under investigation in response to detections of malicious code as a function of the detection routines; and

determining whether code under investigation is malicious code as a function of the weights associated by the detection routines.

96. The method of claim 95, wherein the applying comprises:

applying the detection routines to gather information about the code under investigation by at least one of the following: examining the code under investigation; and searching for information in the information handling system about the code under investigation.

97. The method of claim 95, wherein determining whether the code under investigation is malicious code includes scoring the execution of the detection routines as a function of the weights, wherein scoring includes configuring a scoring algorithm to identify the code under investigation as malicious code in response to a malicious code score.

98. The method of claim 95, wherein the malicious code includes a trojan horse.

99. The method of claim 95, wherein the malicious code includes remote control software.

100. The method of claim 95, wher[illegible]ıe malicious code includes a keystroke logger.

101. The method of claim 95, wherein the malicious code includes spyware.

102. The method of claim 95, wherein the malicious code includes a worm.

103. The method of claim 95, wherein the malicious code includes a virus.

104. The method of claim 95, wherein the malicious code includes monitoring software.

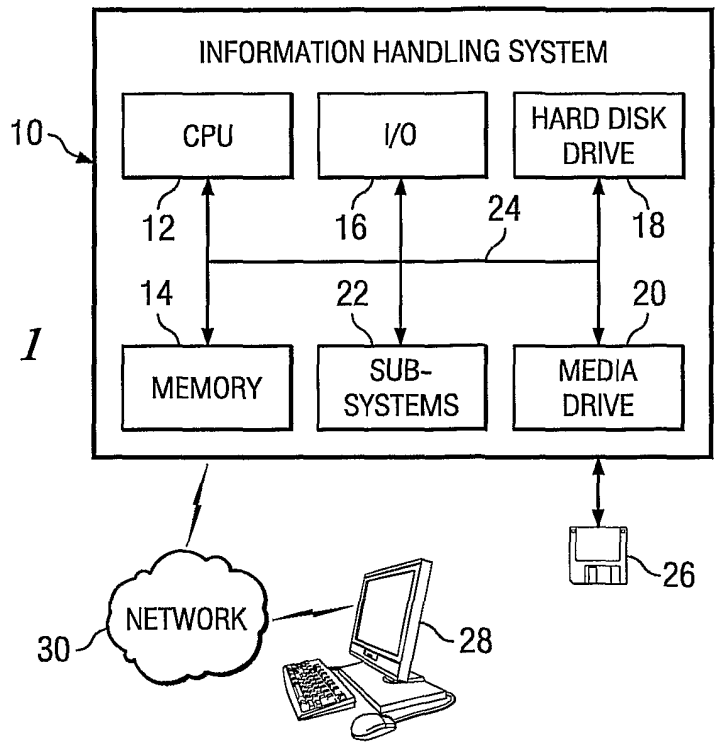

INFORMATION HANDLING SYSTEM
10
CPU
I/O
HARD DISK DRIVE
12
16
24
18
14
22
20
MEMORY
SUB-SYSTEMS
MEDIA DRIVE
26
NETWORK
30
28

Fig. 1

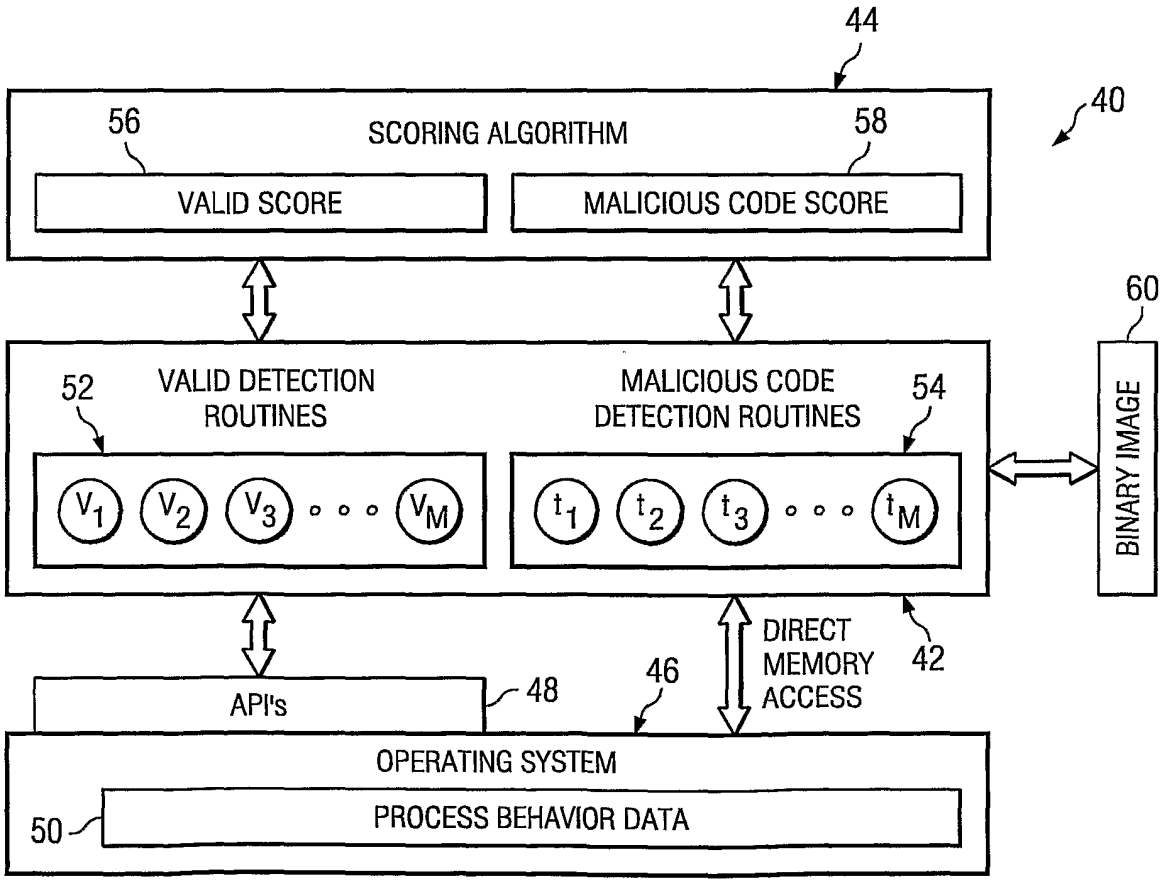

44
40
56
SCORING ALGORITHM
58
VALID SCORE
MALICIOUS CODE SCORE
60
52
VALID DETECTION ROUTINES
MALICIOUS CODE DETECTION ROUTINES
54
V_1 V_2 V_3 ... V_M
t_1 t_2 t_3 ... t_M
BINARY IMAGE
DIRECT MEMORY ACCESS
42
46
API's
48
OPERATING SYSTEM
50
PROCESS BEHAVIOR DATA

Fig. 2

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US03/26993

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : G 06F 12/14; G06F 11/30
US CL : 713/200,201,188

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)
U.S. : 713/200,201,188

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 practicable, search terms used)
IEEE

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 6,393,568 B1 (RANGER et al) 21 May 2002 (21.05.2002), column 4, lines 25-35 and Figs. 1 and 3.	1 - 104
A	US 5,802,277 (COWLARD) 1 September 1998 (1.09.1998), colum 2, lines 35-50; column 6, lines 30-47 and Figs.3 and 4.	1 - 104
A,P	US 2003/0033536 A1 (PAK et al) 13 February 2003 (13.02.2003), page 2, lines 0017; page 6, lines 0115 and Fig. 4.	1 - 104
A	US 5,537,540 (MILLER et al) 16 July 1996 (16.07.1996), column 2, lines 50-67 to column 3, lines 1-4.	1 - 104
A,P	WO 02/103533 A1 (HOEFELMEYER et al) 27 December 2002 (27.12.2002), abstract and Fig. 2	1 - 104

☐ Further documents are listed in the continuation of Box C. ☐ See patent family 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 application or patent 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
11 December 2003 (11.12.2003)

Date of mailing of the international search report
03 FEB 2004

Name and mailing address of the ISA/US
Mail Stop PCT, Attn: ISA/US
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
Facsimile No. (703)305-3230

Authorized officer
Gilberto Barron
Peggy Harwood

Telephone No. (703) 305-3900

Form PCT/ISA/210 (second sheet) (July 1998)