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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
13 March 2014 (13.03.2014)

WIPO | PCT

(10) International Publication Number
WO 2014/039811 A1

(51) **International Patent Classification**:
G06F 21/50 (2013.01) ***G06F 11/30*** (2006.01)

(21) **International Application Number**:
PCT/US2013/058496

(22) **International Filing Date**:
6 September 2013 (06.09.2013)

(25) **Filing Language**: English

(26) **Publication Language**: English

(30) **Priority Data**:
13/607,155 7 September 2012 (07.09.2012) US

(71) **Applicant**: **CROWDSTRIKE, INC.** [US/US]; 30251 Golden Lantern, Laguna Niguel, CA 92677-5993 (US).

(72) **Inventor**: **WICHERSKI, Georg**; Wirichsbongardstr. 39, 52062 Aachen (DE).

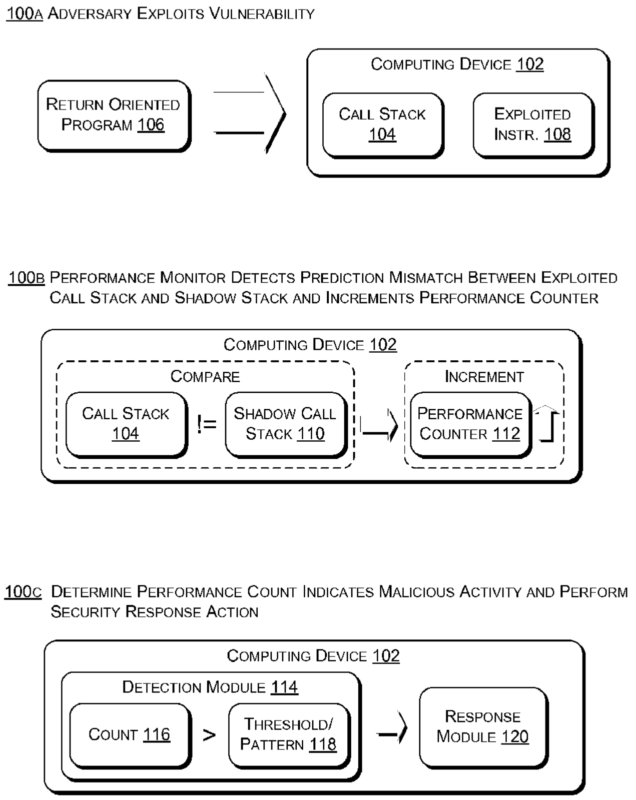
(74) **Agents**: **PECK, Robert, C.** et al.; Lee & Hayes, PLLC, 601 W. Riverside Ave, Suite 1400, Spokane, WA 99201 (US).

(81) **Designated States** *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: THREAT DETECTION FOR RETURN ORIENTED PROGRAMMING


100A Adversary Exploits Vulnerability
Return Oriented Program 106
Computing Device 102
Call Stack 104
Exploited Instr. 108
100B Performance Monitor Detects Prediction Mismatch Between Exploited Call Stack and Shadow Stack and Increments Performance Counter
Computing Device 102
Compare
Call Stack 104
!=
Shadow Call Stack 110
Increment
Performance Counter 112
100C Determine Performance Count Indicates Malicious Activity and Perform Security Response Action
Computing Device 102
Detection Module 114
Count 116
>
Threshold/ Pattern 118
Response Module 120

(57) Abstract: Techniques for detecting security exploits associated with return-oriented programming are described herein. For example, a computing device may determine that a retrieved count is indicative of malicious activity, such as return oriented programming. The computing device may retrieve the count from a processor performance counter of prediction mismatches, the prediction mismatches resulting from comparisons of a call stack of the computing device and of a shadow call stack maintained by a processor of the computing device. In response to determining that the count indicates malicious activity, the computing device may perform at least one security response action.

Published:

— *with international search report (Art. 21(3))*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

THREAT DETECTION FOR RETURN ORIENTED PROGRAMMING

RELATED APPLICATIONS

[0001] This patent application claims priority to U.S. Utility patent application entitled "Threat Detection for Return Oriented Programming" with Serial No. 13/607,155 filed September 7, 2012, which is fully incorporated herein by reference.

BACKGROUND

[0002] With Internet use forming an ever greater part of day to day life, security exploits that steal or destroy system resources, data, and private information are an increasing problem. Governments and businesses devote significant resources to preventing intrusions and thefts related to these security exploits. Security exploits come in many forms, such as computer viruses, worms, trojan horses, spyware, keystroke loggers, adware, rootkits, and shellcodes. These exploits are delivered in or through a number of mechanisms, such as spearfish emails, clickable links, documents, executables, or archives. Some of the threats posed by security exploits are of such significance that they are described as cyber terrorism or industrial espionage.

[0003] A variant of the shellcode security exploits known as Return Oriented Programming (ROP) has proven very difficult to detect. Return oriented programming makes use of a security vulnerability of a computing device to spoof or control the call stack of that computing device. By spoofing or controlling the call stack, the security exploit is able to utilize select instructions of legitimate processes to effectively create and execute a shellcode. The use of legitimate instructions circumvents memory safeguards that have been put in place to stop shellcode security exploits. The only techniques that have been developed for detecting and responding to return oriented programming, however, impose a substantial performance cost.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] The detailed description is set forth with reference to the accompanying figures. In the figures, the left-most digit(s) of a reference number identifies the figure in which the reference number first appears. The use of the same reference numbers in different figures indicates similar or identical items or features.

[0005] FIG. 1 illustrates an overview of techniques for detecting security exploits associated with return oriented programming, the exploits being detected based on a processor performance counter for call stack prediction mismatches.

[0006] FIG. 2 illustrates a component level view of a computing device affected by a return oriented programming security exploit.

[0007] FIG. 3 illustrates an example network connecting an adversary attacking a vulnerability, a computing device affected by that attack, and a remote security service configured to detect and/or respond to the attack.

[0008] FIG. 4 illustrates an example process for retrieving a count of prediction mismatches associated with a call stack, determining that the count is indicative of malicious activity, and, in response, performing at least one security response action.

DETAILED DESCRIPTION

Overview

[0009] This disclosure describes, in part, techniques for detecting security exploits associated with return-oriented programming. The techniques include determining that a retrieved count is indicative of malicious activity, such as return oriented programming. The count may be retrieved from a processor performance counter of prediction mismatches, the prediction mismatches resulting from comparisons of a call stack of a computing device and of a shadow call stack maintained by a processor of the computing device. The techniques further include performing at least one security response action in response to determining that the count indicates malicious activity.

[0010] FIG. 1 illustrates an overview of such techniques. As illustrated in FIG. 1, at 100a, a computing device 102 may have a vulnerability, such as a memory corruption vulnerability, exploited by an adversary. Such a vulnerability may allow an adversary using a return oriented program 106 to control or spoof the call stack 104. By controlling or spoofing the call stack 104, the adversary is able to exploit legitimate instructions 108 (hereinafter “exploited instructions 108”) of one or more processes of the computing device 102 to effectively create and execute a malicious program on the computing device 102. Like shellcode, such a malicious program may be relatively small, even just a few instructions. The return oriented program 106 supplied as part of this security exploit may execute entirely on the affected computing device 102 or may be remotely operated by an adversary system through, for example, a command shell.

[0011] At 100b, a performance monitoring unit associated with a processor of the computing device 102 may predict the value of the call stack 104 using a shadow call stack 110 or other prediction mechanism. For example, the computing device 102 may compare the call stack 104 to a shadow call stack 110 and, if the comparison results in a mismatch, may increment a performance counter 112 of the processor. The computing device 102 maintains a shadow call stack 110 in cache memory of its processor and utilizes the shadow call stack in association with a branch predictor of the computing device 102. The branch predictor attempts to guess which execution path an if-then structure or an indirect branch of a process will take before that path is known. One prediction mechanism used by the branch predictor is the shadow call stack 110, also referred to as a return stack buffer. The performance monitoring unit monitors prediction mismatches from comparisons of the shadow call stack 110 to the call stack 104 stored in system memory of the computing device 102 and, when a prediction mismatch is noted, increments the performance counter 112 specific to prediction mismatches for call stacks. These prediction mismatches may be indicative of malicious activity, such as activity of the return oriented program 106, because the return oriented program 106 may only be able to control or spoof the call stack 104, not the shadow call stack 110. The result of this disparity is often prediction mismatches.

[0012] At 100c, a detection module 114 determines that a count 116 retrieved from the performance counter 112 is indicative of malicious activity. The detection module 114 may do this by comparing the count 116 to a threshold or pattern 118. Such a threshold or pattern 118 may be determined based on monitoring counts 116 of the performance counter 112 over time. The threshold or pattern 118 may also be specific to a process or class of processes, thus allowing processes typically registering prediction mismatches to have different thresholds or pattern 118 and thereby avoiding false positives. For process- or process-class-specific thresholds or pattern 118, the detection module 114 may also retrieve an indication of the process or processes executing at the time that the count 116 was retrieved and then determine an appropriate threshold or pattern 118 based on the indication of the active process(es).

[0013] Also, as shown at 100c, if the detection module 114 determines that the count 116 exceeds the threshold 118 or diverges from the pattern 118, the detection module 114 may invoke or notify the response module 120, the response module 120 performing at least one security response action. For example, the security response action may be providing a graphic, audible, or haptic alert to a user of the computing device 102. Also or instead, the security response action may be notifying a remote security monitoring server of the malicious activity. Further, the security response action may be asking a user of the computing device whether one or more active processes (i.e., the one or more processes being executed by the processor associated with the count 116) should be halted and halting the one or more processes. Additionally, the security response action may be determining information associated with one or more actives processes and analyzing the determined information. The response module 120 may then monitor, or cause another component to monitor, execution activities associated with the one or more active processes.

[0014] In some embodiments, the detection module 114 and the response module 120 may be implemented on the computing device 102. In other embodiments, the detection module 114 and the response module 120 may be implemented by a remote security service. When implementing the detection module 114, the remote security service communicates with logic of the computing device 102 that is configured to retrieve the count 116 from the performance counter 112, enabling the detection module 114 of the remote security service to retrieve the count 116 from that logic. In yet other embodiments, the detection module 114 and the response module 120 may be implemented in part by the computing device 102 and in part by a remote security service.

Example Device

[0015] FIG. 2 illustrates a component level view of a computing device affected by a return oriented programming security exploit. As illustrated, the computing device 102 includes a processor 202, the processor 202 including a performance monitoring unit 204 and cache memory 206. The performance monitoring unit 204 may in turn include one or more performance counters, such as performance counter 112. The cache memory 206 may store the shadow call stack 110.

[0016] The computing device 102 may also include system memory 208. The system memory 208 may store the call stack 104, exploited instructions 108, a security agent 210, the detection module 114, and the response module 120. The response module 120 may in turn include an alert module 212, a report module 214, a remediation module 216, and an analysis module 218.

[0017] In addition, the computing device may include a removable storage 220, non-removable storage 222, input device(s) 224, output device(s) 226 and communication connections 228 for communicating with other computing devices 230.

[0018] In some embodiments, the computing device 102 may be or include a server or server farm, multiple, distributed server farms, a mainframe, a work station, a personal computer (PC), a laptop computer, a tablet computer, a personal digital assistant (PDA), a cellular phone, a media center, an embedded system, or any other sort of device or devices. In one implementation, the computing device 102 represents a plurality of computing devices working in communication, such as a cloud computing network of nodes. In some implementations, the computing device 102 includes one or more virtual machines.

[0019] In various embodiments, the processor 202 is a central processing unit (CPU), such as a processor associated with the 8086 architecture (e.g., the Intel i7® processor) or the 68000 architecture. The computing device may also include one or more other processors, such as a graphic processing unit (GPU), not shown in FIG. 2. In addition to the performance monitoring unit 204 and the cache 206, the processor 202 may include other cache memories, registers, buffers (e.g., translation lookaside buffers), tables, arithmetic logic units (ALUs), interface buses, etc.

[0020] The performance monitoring unit 204 (PMU 204) collects information regarding the performance of the processor 202 and regarding applications or processes being executed by the processor 202. The PMU 204 may include a number of registers and performance counters, the numbers and types of registers and performance counters varying based on the type of the processor 202. Further, the PMU 204 gathers performance information, performs any processing on that information needed to update performance counters, and updates the performance counters. For example, the PMU 204 may compare the call stack 104 to the shadow call stack 110 to determine if there is a prediction mismatch. In some embodiments, this performance information may be obtained, at least in part, from a branch prediction unit of the processor 202. If there is a prediction mismatch, the PMU 204 updates the count for the performance counter 112. While FIG. 2 shows the PMU 204 including the performance counter 112, the PMU 204 may also include other performance counters measuring other aspects of system or process performance.

[0021] The cache 206 may be any sort of cache memory of the processor 202, such as L1 cache or L2 cache. As mentioned above, the cache 206 may store a shadow call stack 110, which is also sometimes referred to as a "return stack buffer." In some embodiments, the shadow call stack 110 may be stored in cache 206 that is even closer to the CPU of processor 202 than the L1 cache. The shadow call stack 110 is used for branch predictions that attempt to predict the state of the call stack 104. In operation, the shadow call stack 110 will often mirror the call stack 104.

[0022] In various embodiments, system memory 208 is volatile (such as RAM), non-volatile (such as ROM, flash memory, etc.) or some combination of the two. As shown, the system memory 208 includes the call stack 104. The call stack 104 is a data structure that stores information about the active subroutines of processes of the computing device 102. For example, the call stack 104 stores the memory address that the subroutine should return control to following operation. As mentioned above, this call stack 104 may be controlled or spoofed by a return oriented program 106 using a vulnerability of the computing device 102 or of one of its applications. By spoofing or controlling the call stack 104, the return oriented program 106 causes control to be returned to the wrong memory addresses. These wrong memory addresses are associated with legitimate, exploited instructions 108 of one or more processes that are then executed in such a manner as to produce malicious activity.

[0023] In various embodiments, the system memory 208 may also include a security agent 210. The security agent 210 may be a kernel-level security agent that observes and acts upon execution activities of the computing device 102. The security agent 210 may be configurable by a remote security service, receiving, and applying while live, reconfigurations of filters, components, models, etc. of the security agent 210. Based on the observed execution activities, the security agents 210 may generate security information which the security agent 210 may act upon and/or provide to the remote security service. While the detection module 114 and response module 120 are shown as being separate from the security agent 210, one or both may, in other embodiments, be components of the security agent 210. An example security agent 210 is described in greater detail in U.S. patent application serial number 13/492,672, entitled “Kernel-Level Security Agent” and filed on June 8, 2012.

[0024] As described above, the detection module 114 may determine a threshold or pattern 118 associated with malicious activity, may retrieve the count 116 from the performance counter 112, and may determine whether the count 116 indicates malicious activity by comparing the count 116 to the threshold or pattern 118. In some embodiments, the detection module 114 may monitor the performance counter 112 over a time period, periodically retrieving its count 116 and synthesizing the retrieved counts 116. These synthesized counts 116 may provide a description of typical values for the performance counter 112, and the detection module 114 may set the threshold 118 or pattern based at least in part on the counts and/or synthesized counts 116. In further embodiments, the detection module 114 may determine a threshold or pattern 118 for each process or each class or type of process by concurrently monitoring the performance counter 112 and active process(es).

[0025] The detection module 114 may further retrieve the count 116 from the performance counter 112, either periodically or in response to the occurrence of one or more triggering events. The processor 202 may include an interface enabling application processes or platform-level processes to obtain the count 116, and the detection module 114 may utilize that interface. Upon retrieving the count 116, the detection module 114 compares the count 116 to the threshold or pattern 118. If the count 116 exceeds the threshold 118 or diverges from the pattern 118, the detection module 114 determines that the count 116 indicates malicious activity and, in response, invokes the response module 120. If the threshold is specific to a process or class or type of processes, the detection module 114 may also obtain an indication of the active process or processes and select an appropriate threshold or pattern 118. If multiple processes associated with different thresholds or patterns 118 are active, the detection module 114 may, for example, select the highest value threshold or pattern 118.

[0026] In various embodiments, the response module 120 may determine an appropriate response to the malicious activity detected by the detection module 114. The response module 120 may include a number of modules associated with varying responses, such as an alert module 212, a report module 214, a remediation module 216, and an analysis module 218. In some embodiments, there may be no response module 120, with the modules 212-218 taking the place of the response module 120 and being invoked directed by the detection module 114. The response module 120 may invoke any one or more of the modules 212-218 in order to respond appropriately to the malicious activity. The module(s) 212-218 invoked may depend on settings or a configuration of the response module 120.

[0027] The alert module 212 may provide the user of the computing device 102 with a visual, audible, or haptic alert of the malicious activity. In some embodiments, the alert is simply informative. In other embodiments, the alert may present the user with one or more options for responding to the malicious activity, such as a report option which may result in invocation of the report module 214 or a remediation option with may result in invocation of the remediation module 216 or of the security agent 210.

[0028] In some embodiments, the report module 214 may prepare and send a report of the malicious activity to a remote security service. The report module 214 may be invoked by the response module 120 or by the alert module 212 responsive to a user selection of a reporting option. The report generated by the report module 214 may include the count 116, the threshold or pattern 118, an indication of one or more active processes, and/or information about the state of the call stack 104 and/or the shadow call stack 110.

[0029] In various embodiments, the remediation module 216 may halt one or more active processes. The remediation module 216 may be invoked by the response module 120 or by the alert module 212 responsive to a user selection of a remediation option. In some embodiments, prior to halting the one or more active processes, the remediation module 216 may ask the user whether the user wishes to halt the one or more active processes. If the user elects to halt the one or more active processes, then the remediation module 216 may halt those process(es).

[0030] In further embodiments, an analysis module 218 may determine information associated with the one or more active processes and may analyze that determined information. For example, if the security agent 210 or other computing device component maintains a model of execution activities of the one or more active processes, the analysis module 218 may retrieve the information associated with the active process(es) and compare it to the model. Such analysis may detect differences in execution flow that may confirm the determination that malicious activity is occurring. The analysis module 218 may then either monitor the execution activities of the one or more active processes or invoke the security agent 210 or other computing device component to perform the monitoring. Such monitoring may enable the computing device 102 to obtain more information about the malicious activity after it has been detected.

[0031] Computing device 102 also includes additional data storage devices (removable and/or non-removable) such as, for example, magnetic disks, optical disks, or tape. Such additional storage is illustrated in FIG. 2 by removable storage 220 and non-removable storage 222. Tangible computer-readable media may include volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information, such as computer readable instructions, data structures, program modules, or other data. System memory 208, removable storage 218 and non-removable storage 220 are all examples of tangible computer-readable media. Tangible computer-readable media include, but are not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other tangible medium which can be used to store the desired information and which can be accessed by the computing device 102. Any such tangible computer-readable media may be part of the computing device 102.

[0032] Computing device 102 also has input device(s) 224, such as a keyboard, a mouse, a touch-sensitive display, voice input device, etc., and output device(s) 226 such as a display, speakers, a printer, etc. These devices are well known in the art and need not be discussed at length here.

[0033] Computing device 102 also contains communication connections 228 that allow the computing device 102 to communicate with other computing devices 230, such as a remote security service or an adversary system.

Example Network

[0034] FIG. 3 illustrates an example network 302 connecting an adversary 304 exploiting a vulnerability with a return oriented program 106, a computing device 102 affected by that security exploit, and a remote security service 306 configured to detect, monitor, and/or respond to the security exploit.

[0035] In some embodiments, the network 302 may include any one or more networks, such as wired networks, wireless networks, and combinations of wired and wireless networks. Further, the network 302 may include any one or combination of multiple different types of public or private networks (e.g., cable networks, the Internet, wireless networks, etc.). In some instances, computing devices communicate over the network 302 using a secure protocol (e.g., https) and/or any other protocol or set of protocols, such as the transmission control protocol/Internet protocol (TCP/IP).

[0036] In various embodiments, the adversary system 304 and the remote security service 306 may each be or include a server or server farm, multiple, distributed server farms, a mainframe, a work station, a personal computer (PC), a laptop computer, a tablet computer, a personal digital assistant (PDA), a cellular phone, a media center, an embedded system, or any other sort of device or devices. In one implementation, the computing devices of the remote security service 306 represent a plurality of computing devices working in communication, such as a cloud computing network of nodes. When implemented on multiple computing devices, the remote security service 306 may distribute the detection module 114 and response module 120 among the multiple computing devices. In some implementations, one or more of the adversary system 304 and remote security service 306 represent one or more virtual machines implemented on one or more computing devices.

[0037] In some embodiments, the adversary system 304 may be any computing device configured to utilize a return oriented program 106 to exploit a vulnerability, such as a memory corruption vulnerability that enables an adversary system 304 to control or spoof a call stack 104. As mentioned above, the return oriented program 106 may execute entirely on the computing device 102 or may be remotely controlled through the adversary system 304. Such remote control may involve a command shell or other interface provided by the adversary system 304 to its adversary user.

[0038] In various embodiments, the remote security service 306 may provide monitoring, configuration and healing services to the computing device 102. Such services may include, for example, configuring or reconfiguring the security agent 210, installing the security agent 210, receiving reports and alerts from computing devices, and/or responding to an alert or report with healing, agent reconfiguration, or further monitoring. In some embodiments, as shown, the remote security service 306 may include part or all of one or both of the detection module 114 and the response module 120 and may execute that module or those modules in the manner described above. When implementing the detection module 114, the remote security service 306 communicates with logic of the computing device 102 that is configured to retrieve the count 116 from the performance counter 112, enabling the detection module 114 of the remote security service 306 to retrieve the count 116 from that logic. An example of such a remote security service 306 is described in greater detail in U.S. patent application serial number 13/492,672, entitled “Kernel-Level Security Agent” and filed on June 8, 2012.

[0039] In further embodiments, the remote security service 306 may provide a collaboration service that connects multiple client computing devices 102 associated with a same entity or with different entities. Such a collaboration service may relay an alert or report received from one computing device 102 to other computing devices 102, or may generate a new configuration or monitoring process to apply across a group based on an alert or report from one group member. An example remote security service 306 offering such a collaboration service is described in greater detail in U.S. patent application serial number 13/538,439, entitled “Social Sharing of Security Information in a Group” and filed on June 29, 2012.

Example Processes

[0040] FIG. 4 illustrates an example process 400. This process is illustrated as a logical flow graph, each operation of which represents a sequence of operations that can be implemented in hardware, software, or a combination thereof. In the context of software, the operations represent computer-executable instructions stored on one or more computer-readable storage media that, when executed by one or more processors, perform the recited operations. Generally, computer-executable instructions include routines, programs, objects, components, data structures, and the like that perform particular functions or implement particular abstract data types. The order in which the operations are described is not intended to be construed as a limitation, and any number of the described operations can be combined in any order and/or in parallel to implement the processes.

[0041] FIG. 4 illustrates an example process for retrieving a count of prediction mismatches associated with a call stack, determining that the count is indicative of malicious activity, and, in response, performing at least one security response action. The process 400 includes, at 402, determining a prediction mismatch threshold or pattern. The prediction mismatch threshold or pattern may be indicative of a number or pattern of prediction mismatches expected to arise from comparisons of the call stack of a computing device with the shadow call stack implemented in a cache memory of a processor of the computing device. A computing device may, for example, determine the threshold or pattern by monitoring, over time, counts of a processor performance counter for prediction mismatches. Also, in some embodiments, the computing device may utilize different thresholds or pattern for different processes or classes of processes.

[0042] At 404, the computing device may retrieve the count of prediction mismatches from the processor performance counter. At 406, the computing device may then determine whether the count is indicative of malicious activity, such as return oriented programming, based at least in part on a comparison of the count to the threshold or pattern.

[0043] At 408, in response to determining that the count is indicative of malicious activity, the computing device may determine one or more security response actions. At 410, those security response actions may include providing a graphic, audible, or haptic alert to a user of the computing device. At 412, the security response actions may include notifying a remote security monitoring server of the malicious activity. At 414-416, the security response actions may include asking (at 414) a user of the computing device whether the one or more processes should be halted and halting (at 416), the one or more processes. At 418-422, the security responses may include determining (at 418) information associated with one or more processes and analyzing (at 420) the determined information. At 422, the computing device may then monitor execution activities associated with the one or more processes.

[0044] In various embodiments, the operations shown at blocks 402-422 may be performed by the computing device affected by the malicious activity, by a remote security service, or partly by each of the computing device and remote security service.

CONCLUSION

[0045] Although the subject matter has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the specific features or acts described. Rather, the specific features and acts are disclosed as exemplary forms of implementing the claims.

CLAIMS

WHAT IS CLAIMED IS:

1. A computer-implemented method comprising:

retrieving a count of prediction mismatches from a processor performance counter, the prediction mismatches resulting from comparisons of a call stack of a computing device and of a shadow call stack maintained by a processor of the computing device;

determining whether the count indicates malicious activity; and

in response to determining that the count indicates malicious activity, performing at least one security response action.

2. The method of claim 1, wherein the determining whether the count indicates malicious activity comprises determining whether the count exceeds a threshold or diverges from a pattern.

3. The method of claim 2, wherein the threshold or the pattern may be specific to a process or class of processes.

4. The method of claim 2, further comprising determining the threshold or the pattern based on monitoring the count over a time period.

5. The method of claim 1, wherein the at least one security response action is providing a graphic, audible, or haptic alert to a user of the computing device.

6. The method of claim 1, wherein the at least one security response action is notifying a remote security monitoring server of the malicious activity.

7. The method of claim 1, wherein the at least one security response action is halting execution of one or more processes.

8. The method of claim 1, further comprising, prior to halting, asking a user of the computer device whether the one or more processes should be halted and performing the halting conditionally on a user response.

9. The method of claim 1, wherein the at least one security response action is determining information associated with one or more processes and analyzing the determined information.

10. The method of claim 9, further comprising, after determining whether the count indicates malicious activity, monitoring execution activities associated with the one or more processes.

11. The method of claim 1, wherein the retrieving, determining, and performing are performed by the computing device, by a remote security service, or in part by both of the computing device and the remote security service.

12. One or more tangible computer-readable media storing computer-executable instructions configured to program one or more computing devices to perform operations comprising:

retrieving a count of prediction mismatches from a processor performance counter, the prediction mismatches resulting from comparisons of a call stack of a computing device and of a shadow call stack maintained by a processor of the computing device;

determining whether the count indicates malicious activity; and

in response to determining that the count indicates malicious activity, performing at least one security response action.

13. The one or more tangible computer-readable media of claim 12, wherein the computing device is one of the one or more computing devices.

14. The one or more tangible computer-readable media of claim 12, wherein the at least one security response action is providing a graphic, audible, or haptic alert to a user of the computing device.

15. The one or more tangible computer-readable media of claim 12, wherein the at least one security response action is notifying a remote security monitoring server of the malicious activity.

16. The one or more tangible computer-readable media of claim 12, wherein the at least one security response action is halting execution of one or more processes.

17. The one or more tangible computer-readable media of claim 12, wherein the at least one security response action is determining information associated with one or more processes and analyzing the determined information.

18. A computing device comprising:

a processor;

a detection module configured to be operated by the processor to

retrieve a count of prediction mismatches from a processor performance counter, the prediction mismatches resulting from comparisons of a call stack of the computing device and of a shadow call stack maintained by the processor, and

determine whether the count indicates malicious activity; and a response module configured to be operated by the processor to perform, in response to determining that the count indicates malicious activity, at least one of:

providing a graphic, audible, or haptic alert to a user of the computing device,

notifying a remote security monitoring server of the malicious activity,

halting execution of one or more processes, or

determining information associated with the one or more processes and analyzing the determined information.

19. The computing device of claim 18, wherein the detection module is configured to determine whether the count indicates malicious activity by determining whether the count exceeds a threshold or diverges from a pattern, and the threshold or the pattern is based on monitoring the count over a time period.

20. The computing device of claim 18, further comprising an agent configured to monitor execution activities associated with the one or more processes after determining whether the count indicates malicious activity.

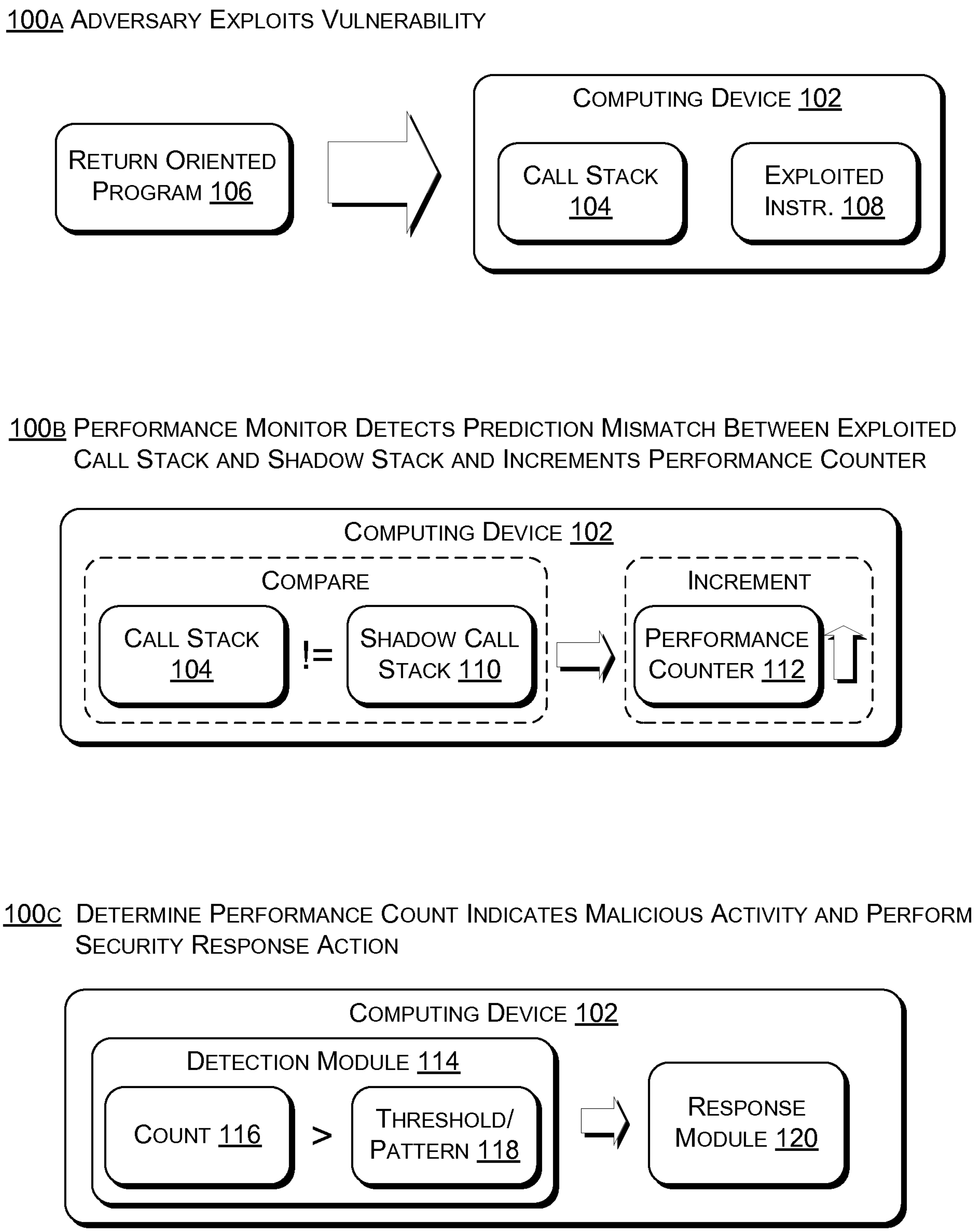

100A ADVERSARY EXPLOITS VULNERABILITY
RETURN ORIENTED PROGRAM 106
COMPUTING DEVICE 102
CALL STACK 104
EXPLOITED INSTR. 108
100B PERFORMANCE MONITOR DETECTS PREDICTION MISMATCH BETWEEN EXPLOITED CALL STACK AND SHADOW STACK AND INCREMENTS PERFORMANCE COUNTER
COMPUTING DEVICE 102
COMPARE
CALL STACK 104
!=
SHADOW CALL STACK 110
INCREMENT
PERFORMANCE COUNTER 112
100C DETERMINE PERFORMANCE COUNT INDICATES MALICIOUS ACTIVITY AND PERFORM SECURITY RESPONSE ACTION
COMPUTING DEVICE 102
DETECTION MODULE 114
COUNT 116
>
THRESHOLD/ PATTERN 118
RESPONSE MODULE 120

FIG. 1

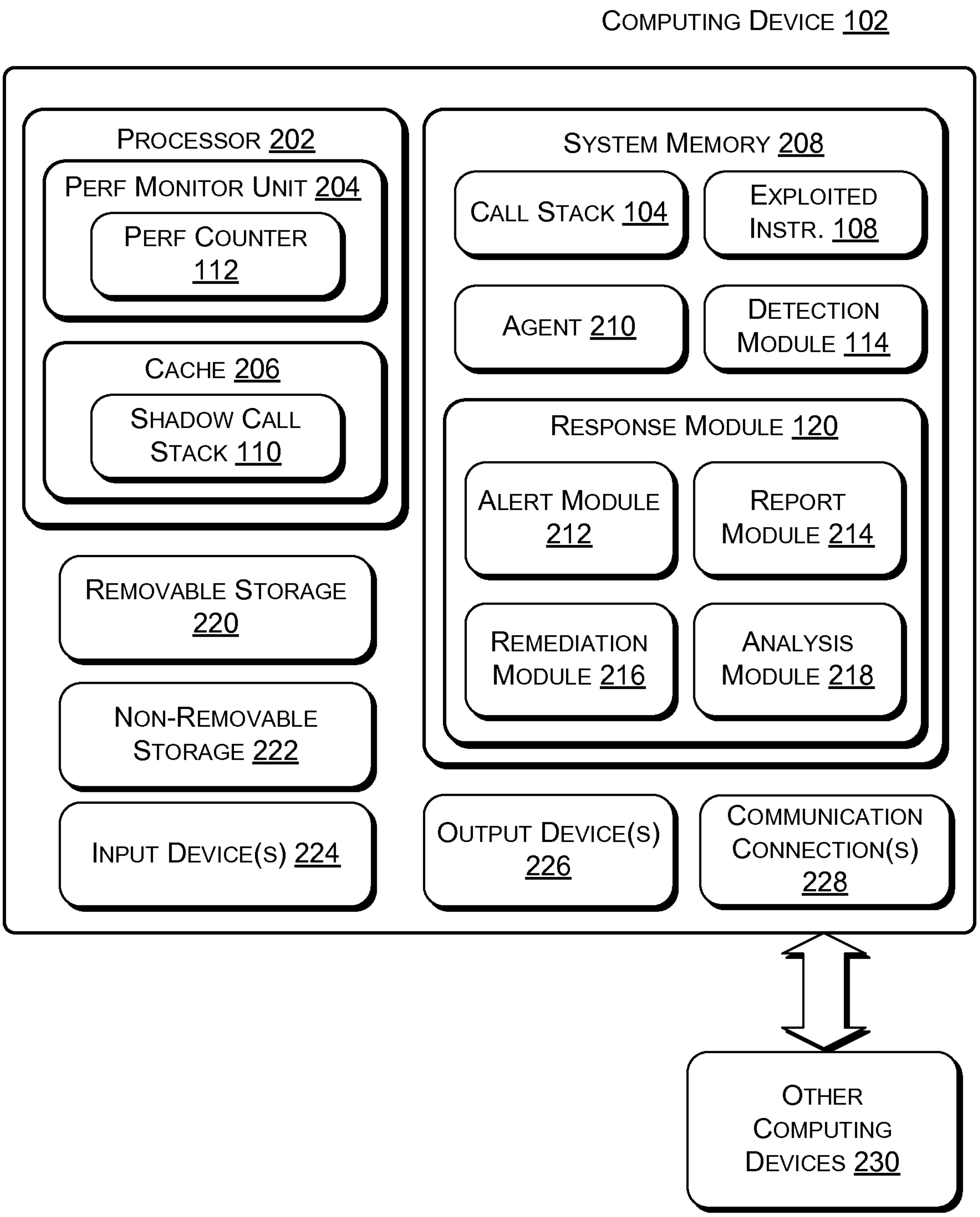

COMPUTING DEVICE 102
PROCESSOR 202
PERF MONITOR UNIT 204
PERF COUNTER 112
CACHE 206
SHADOW CALL STACK 110
SYSTEM MEMORY 208
CALL STACK 104
EXPLOITED INSTR. 108
AGENT 210
DETECTION MODULE 114
RESPONSE MODULE 120
ALERT MODULE 212
REPORT MODULE 214
REMEDIATION MODULE 216
ANALYSIS MODULE 218
REMOVABLE STORAGE 220
NON-REMOVABLE STORAGE 222
INPUT DEVICE(S) 224
OUTPUT DEVICE(S) 226
COMMUNICATION CONNECTION(S) 228
OTHER COMPUTING DEVICES 230

FIG. 2

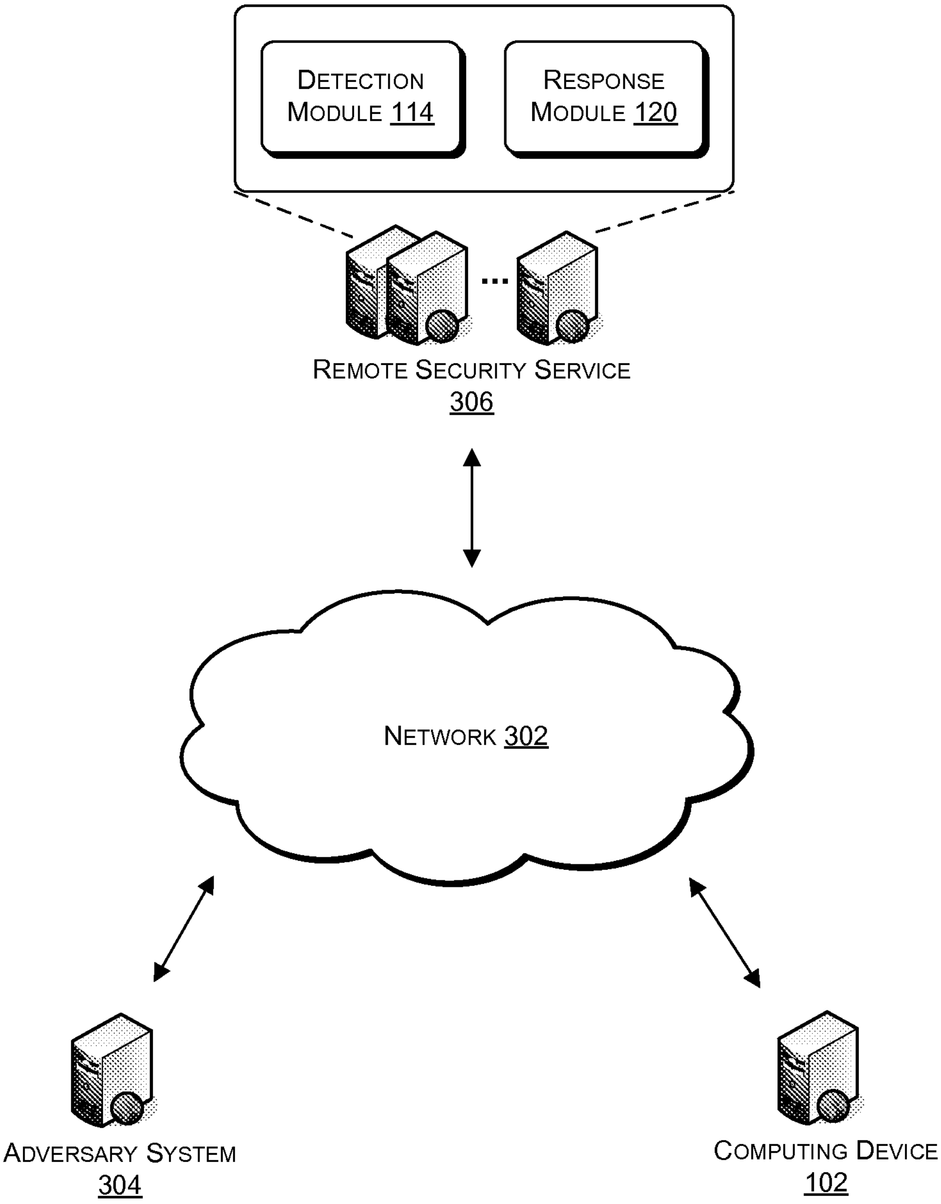

DETECTION MODULE 114
RESPONSE MODULE 120
REMOTE SECURITY SERVICE 306
NETWORK 302
ADVERSARY SYSTEM 304
COMPUTING DEVICE 102

FIG. 3

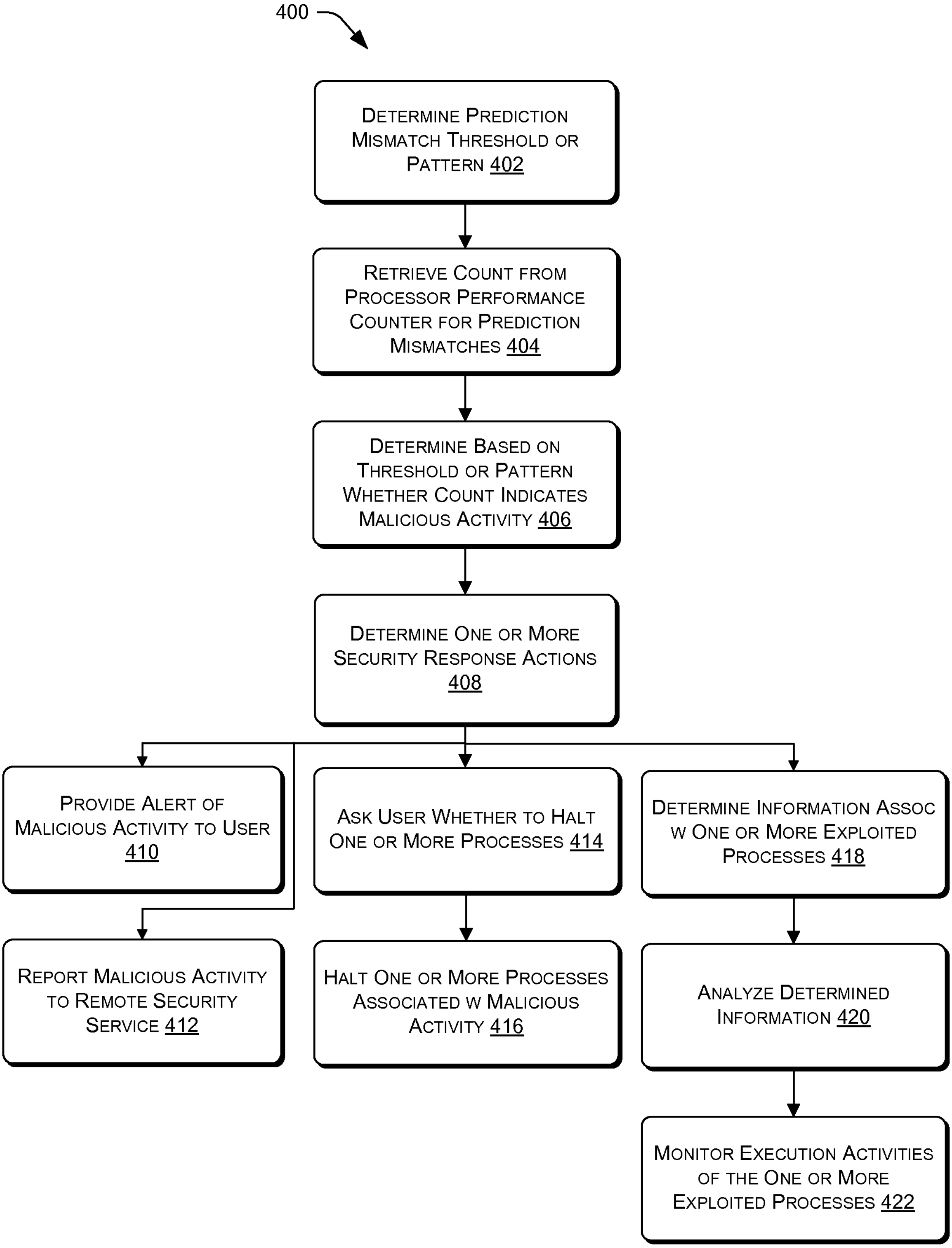

400
Determine Prediction Mismatch Threshold or Pattern 402
Retrieve Count from Processor Performance Counter for Prediction Mismatches 404
Determine Based on Threshold or Pattern Whether Count Indicates Malicious Activity 406
Determine One or More Security Response Actions 408
Provide Alert of Malicious Activity to User 410
Ask User Whether to Halt One or More Processes 414
Determine Information Assoc w One or More Exploited Processes 418
Report Malicious Activity to Remote Security Service 412
Halt One or More Processes Associated w Malicious Activity 416
Analyze Determined Information 420
Monitor Execution Activities of the One or More Exploited Processes 422

FIG. 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2013/058496

A. CLASSIFICATION OF SUBJECT MATTER

G06F 21/50(2013.01)i, G06F 11/30(2006.01)i

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)
G06F 21/50; G06F 12/14; G06F 9/00; G06F 9/38; G06F 11/30

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: Return Oriented Programming, shellcode, detect, monitor, analyze, malicious, shadow call stack, call stack, processor performance counter, mismatch, threshold, pattern, count, remote, service

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	LUCAS DAVI et al., `ROPdefender: A Detection Tool to Defend Against Return-Oriented Programming Attacks`, In: Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security, Hong Kong, China, 22-24 Mahch 2011, Pages 40-51 See pages 40, 42-43, 46.	1-20
Y	LIWEI YUAN et al., `Security Breaches as PMU Deviation: Detecting and Identifying Security Attacks Using Performance Counters`, In: Proceedings of the Second Asia-Pacific Workshop on Systems, Shanghai, China, 11-12 July 2011 , Aiticle No. 6 See pages 1-4.	1-20
A	US 2009-0320129 A1 (AIMIN PAN et al.) 24 December 2009 See paragraphs [0001]-[0015], [0030]-[0033], [0036]-[0038], [0050]-[0058], [0065]-[0066]; and claims 1-2.	1-20
A	KEITH A. BARE, `CPU Performance Counter-Based Problem Diagnosis for Software Systems`, September 2009, Retrieved from http://reports-archive.adm.cs.cmu.edu/anon/2009/CMU-CS-09-158.pdf See pages 1-3, 5-6, 23.	1-20

☒ 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 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 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
26 December 2013 (26.12.2013)

Date of mailing of the international search report
27 December 2013 (27.12.2013)

Name and mailing address of the ISA/KR
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City, 302-701, Republic of Korea
Facsimile No. +82-42-472-7140

Authorized officer
BYUN, Sung Cheal
Telephone No. +82-42-481-8262

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2004-0143727 A1 (THOMAS C. MCDONALD) 22 July 2004 See paragraphs [0002]-[0013], [0019]-[0020], [0023]-[0027], [0029]-[0037], [0039]-[0040], [0043], [0062], [0068]-[0074]; and claims 1, 7-8, 10-14.	1-20
A	US 2009-0089564 A1 (ERNIE F. BRICKELL et al.) 02 April 2009 See paragraphs [0002]-[0008], [0014]-[0033], [0038]; and claim 1.	1-20

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2013/058496

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2009-0320129 A1	24/12/2009	US 8117660 B2	14/02/2012
US 2004-0143727 A1	22/07/2004	TW 282514 B TW I282514 B US 7178010 B2	11/06/2007 11/06/2007 13/02/2007
US 2009-0089564 A1	02/04/2009	None	