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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
2 July 2015 (02.07.2015)

WIPO | PCT

(10) International Publication Number
WO 2015/100388 A1

(51) International Patent Classification:
*G06F 21/56* (2013.01) *H04L 29/06* (2006.01)

(21) International Application Number:
PCT/US2014/072292

(22) International Filing Date:
23 December 2014 (23.12.2014)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/921,033 26 December 2013 (26.12.2013) US
14/228,073 27 March 2014 (27.03.2014) US

(71) Applicant: FIREEYE, INC. [US/US]; 1440 McCarthy Boulevard, Milpitas, California 95035 (US).

(72) Inventors: AZIZ, Ashar; 1440 McCarthy Boulevard, Milpitas, California 95035 (US). BU, Zheng; 1440 McCarthy Boulevard, Milpitas, California 95035 (US). AMIN, Muhammad; 1440 McCarthy Boulevard, Milpitas, California 95035 (US). ISMAEL, Osman Abdoul; 1440 McCarthy Boulevard, Milpitas, California 95035 (US).

(74) Agents: SCHAAL, William W. et al.; Blakely Sokoloff Taylor & Zafman LLP, 1279 Oakmead Parkway, Sunnyvale, California 94085 (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, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, 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.

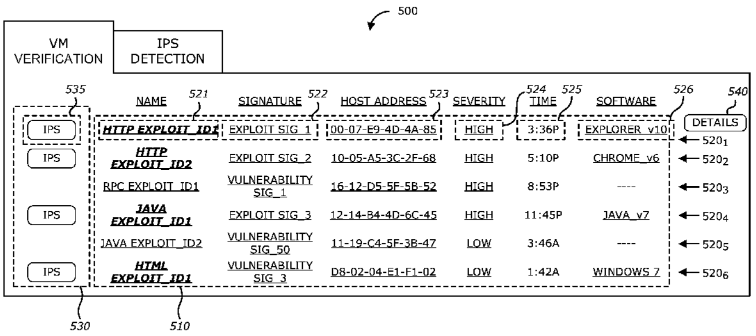
(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, 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).

**Published**:

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

— *with amended claims (Art. 19(1))*

# (54) Title: SYSTEM, APPARATUS AND METHOD FOR AUTOMATICALLY VERIFYING EXPLOITS WITHIN SUSPECT OBJECTS AND HIGHLIGHTING THE DISPLAY INFORMATION ASSOCIATED WITH THE VERIFIED EXPLOITS



| NAME | SIGNATURE | HOST ADDRESS | SEVERITY | TIME | SOFTWARE |
|---|---|---|---|---|---|
| ***HTTP EXPLOIT_ID1*** | EXPLOIT SIG_1 | 00-07-E9-4D-4A-85 | HIGH | 3:36P | EXPLORER_v10 |
| ***HTTP EXPLOIT_ID2*** | EXPLOIT SIG_2 | 10-05-A5-3C-2F-68 | HIGH | 5:10P | CHROME_v6 |
| RPC EXPLOIT_ID1 | VULNERABILITY SIG_1 | 16-12-D5-5F-5B-52 | HIGH | 8:53P | ---- |
| ***JAVA EXPLOIT_ID1*** | EXPLOIT SIG_3 | 12-14-B4-4D-6C-45 | HIGH | 11:45P | JAVA_v7 |
| JAVA EXPLOIT_ID2 | VULNERABILITY SIG_50 | 11-19-C4-5F-3B-47 | LOW | 3:46A | ---- |
| ***HTML EXPLOIT_ID1*** | VULNERABILITY SIG_3 | D8-02-04-E1-F1-02 | LOW | 1:42A | WINDOWS 7 |

***FIG. 5A***

(57) **Abstract**: According to one embodiment, a threat detection system is integrated with intrusion protection system (IPS) logic and virtual execution logic. The IPS logic is configured to receive a first plurality of objects and filter the first plurality of objects by identifying a second plurality of objects as suspicious objects. The second plurality of objects is a subset of the first plurality of objects and is lesser or equal in number to the first plurality of objects. The virtual execution logic is configured to automatically verify whether any of the suspicious objects is an exploit. The virtual execution logic comprises at least one virtual machine configured to virtually process content within the suspicious objects and monitor for anomalous behaviors during the virtual processing that are indicative of exploits.

# SYSTEM, APPARATUS AND METHOD FOR AUTOMATICALLY VERIFYING EXPLOITS WITHIN SUSPECT OBJECTS AND HIGHLIGHTING THE DISPLAY INFORMATION ASSOCIATED WITH THE VERIFIED EXPLOITS

## CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims the benefit of priority on U.S. Provisional Application No. 61/921,033, filed December 26, 2013, the entire contents of which are incorporated by reference herein.

## 1. Field

Embodiments of the disclosure relate to the field of network security. More specifically, one embodiment of the disclosure relates to a system, apparatus and method for identifying a suspicious object, automatically verifying the suspect object as an exploit through virtual processing.

## General Background

Over the last decade, malicious software has become a pervasive problem for Internet users as most networked resources include software vulnerabilities that are subject to attack. For instance, over the past few years, more and more vulnerabilities are being discovered in software that is loaded onto network devices, such as vulnerabilities within operating systems for example. While some vulnerabilities continue to be addressed through software patches, prior to the release of such software patches, network resources continue to be the targeted by exploits.

In general, an exploit is information that attempts to take advantage of a vulnerability in computer software by adversely influencing or attacking normal operations of a targeted computer. As an illustrative example, a Portable Execution Format (PDF) file may be infected with an exploit that is activated upon execution (opening) of the PDF file and takes advantage of a vulnerability associated with Acrobat® Reader version 9.0.

Currently, one type of security application widely used for detecting exploits is an intrusion prevention system (IPS). Typically implemented as part of a firewall, an IPS is designed to identify packets suspected of containing known exploits, attempt to block/halt propagation of such exploits, and log/report information associated with such packets through an alert. However, conventional IPS technology suffers from a number of disadvantages.

One disadvantage with conventional IPS technology in that the IPS does not rely on any mechanism to automatically verify its results. Rather, verification of the results produced from a conventional IPS is handled manually.

Another disadvantage is that, without automated verification, the IPS tends to produce a large number of false positives, namely incorrect alerts that occur when the IPS reports certain benign objects as exploits. These false positives cause a variety of adverse effects. For instance, due to the large number of false positives, one adverse effect is that actual exploits detected within network traffic may go unnoticed by an administrator. Other adverse effects may include (i) needless blocking of incoming network traffic; (ii) unnecessarily reduction of processing resources; (iii) significant drainage of administrative resources to handle incorrectly classified objects; and (iv) development of a culture (or policy) of sporadically checking only some of the suspect objects.

In efforts to mitigate the number of false positives, the IPS may frequently require customized and periodic tuning of its signature database, which is a costly endeavor. Furthermore, simply tuning the IPS to significantly reduce the number of false positives can severely degrade the effectiveness of the IPS and/or severely disrupt network operability.

## BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the invention are illustrated by way of example and not by way of limitation in the figures of the accompanying drawings, in which like references indicate similar elements and in which:

FIG. 1A is a first exemplary block diagram of an operational flow of threat detection and prevention within an electronic device.

FIG. 1B is a second exemplary block diagram of an operational flow of threat detection and prevention within an electronic device.

FIG. 2A is a first exemplary block diagram of a communication system deploying a plurality of threat detection and prevention (TDP) systems with framework for conducting exploit analysis using intrusion protection system (IPS) logic with results verified by virtual execution logic.

FIG. 2B is a second exemplary block diagram of a communication system deploying a plurality of TDP systems with framework for conducting exploit analysis using IPS logic with results verified by virtual execution logic.

FIG. 3 is an exemplary block diagram of logic associated with the TDP system of FIGs. 2A-2B.

FIG. 4 is an exemplary diagram of a flowchart illustrating operations of the threat detection and prevention process.

FIGs. 5A-5B are exemplary embodiments of user interface display screens produced by display logic, where the display screens provides an interactive dashboard.

FIGs. 6A-6B are exemplary block diagrams of operational flows of analyzed objects associated with network traffic in accordance with an alternative embodiment of the TCP system.

FIGs. 7A-7B are exemplary block diagrams of a communication system deploying a plurality of threat detection and prevention (TDP) systems with a framework for conducting exploit analysis using intrusion protection system (IPS) logic and heuristic logic with results verified by the virtual execution logic pursuant to the alternative embodiment.

FIGs. 8A-8B are exemplary diagrams of a flowchart illustrating operations of the threat detection and prevention process according to the framework of FIGs. 7A-7B.

## DETAILED DESCRIPTION

Various embodiments of the disclosure relate to an electronic device with network connectivity, such as a threat detection and prevention (TDP) system for example, where the electronic device comprises a static analysis engine, a dynamic analysis engine and reporting logic. According to one embodiment of the disclosure, the static analysis engine comprises intrusion protection system (IPS) logic that conducts at least exploit signature checks and/or vulnerability signature checks on objects under analysis to identify whether characteristics of any of these objects are indicative of an exploit. Those objects with these identified characteristics are label "suspect" or "suspicious" objects. The dynamic analysis engine comprises virtual execution logic to automatically and subsequently analyze, without user assistance, content within suspect objects provided from the IPS logic in order to possibly verify whether any of the suspect objects is an exploit.

Based on analysis results from the IPS logic and the virtual execution logic, reporting logic within the TDP system generates a report (e.g., one or more display screens, printed report, etc.) that highlights information associated with these "verified" exploits, namely suspect objects initially identified by the IPS logic that have been verified by the virtual execution logic to be exploits. Some or all of the information associated with the verified exploits (referred to as "verified exploit information") may be highlighted to visibly denote the verified exploits from the non-verified exploits, namely suspect objects initially identified by the IPS logic that have not been verified by the virtual execution logic. Examples as to how the verified exploit information is highlighted may include (1) altering location or ordering of at least certain portions of the verified exploit information to prominently display such information within the report; (2) modifying the font (e.g., color, size, type, style, and/or effects) used in conveying some of the verified exploit information; (3) placement of one or more images proximate to a listing of the verified exploit information; or the like.

## Terminology

In the following description, certain terminology is used to describe features of the invention. For example, in certain situations, both terms "logic" and "engine" are representative of hardware, firmware and/or software that is configured to perform one or more functions. As hardware, logic (or engine) may include circuitry having data processing or storage functionality. Examples of such circuitry may include, but is not limited or restricted to a microprocessor, one or more processor cores, a programmable gate array, a microcontroller, an application specific integrated circuit, wireless receiver, transmitter and/or transceiver circuitry, semiconductor memory, or combinatorial logic.

Logic (or engine) may be software in the form of one or more software modules, such as executable code in the form of an executable application, an application programming interface (API), a subroutine, a function, a procedure, an applet, a servlet, a routine, source code, object code, a shared library/dynamic load library, or one or more instructions. These software modules may be stored in any type of a suitable non-transitory storage medium, or transitory storage medium (e.g., electrical, optical, acoustical or other form of propagated signals such as carrier waves, infrared signals, or digital signals). Examples of non-transitory storage medium may include, but are not limited or restricted to a programmable circuit; a semiconductor memory; non-persistent storage such as volatile memory (e.g., any type of random access memory "RAM"); persistent storage such as non-volatile memory (e.g., read-only memory "ROM", power-backed RAM, flash memory, phase-change memory, etc.), a solid-state drive, hard disk drive, an optical disc drive, or a portable memory device. As firmware, the executable code is stored in persistent storage.

The term "object" generally refers to a collection of data, whether in transit (e.g., over a network) or at rest (e.g., stored), often having a logical structure or organization that enables it to be classified for purposes of analysis. During analysis, for example, the object may exhibit a set of expected characteristics and, during processing, a set of expected behaviors. The object may also exhibit a set of unexpected characteristics and a set of unexpected behaviors that may evidence an exploit and potentially allow the object to be classified as an exploit.

Examples of objects may include one or more flows or a self-contained element within a flow itself. A "flow" generally refers to related packets that are received, transmitted, or exchanged within a communication session. For convenience, a packet is broadly referred to as a series of bits or bytes having a prescribed format, which may include packets, frames, or cells.

As an illustrative example, an object may include a set of flows such as (1) a sequence of transmissions in accordance with a particular communication protocol (e.g., User Datagram Protocol (UDP); Transmission Control Protocol (TCP); or Hypertext Transfer Protocol (HTTP); etc.), or (2) inter-process communications (e.g. Remote Procedure Call "RPC" or analogous processes, etc.). Similar, as another illustrative example, the object may be a self-contained element, where different types of such objects may include an executable file, non-executable file (such as a document or a dynamically link library), a Portable Document Format (PDF) file, a JavaScript file, Zip file, a Flash file, a document (for example, a Microsoft Office® document), an electronic mail (email), downloaded web page, an instant messaging element in accordance with Session Initiation Protocol (SIP) or another messaging protocol, or the like.

An “exploit” may be construed broadly as information (e.g., executable code, data, command(s), etc.) that attempts to take advantage of a software vulnerability. Typically, a “vulnerability” is a coding error or artifact of software (e.g., computer program) that allows an attacker to alter legitimate control flow during processing of the software (computer program) by an electronic device, and thus, causes the electronic device to experience undesirable or unexpected behaviors. The undesired or unexpected behaviors may include a communication-based anomaly or an execution-based anomaly, which, for example, could (1) alter the functionality of an electronic device executing application software in a malicious manner; (2) alter the functionality of the electronic device executing that application software without any malicious intent; and/or (3) provide unwanted functionality which may be generally acceptable in another context. To illustrate, a computer program may be considered as a state machine, where all valid states (and transitions between states) are managed and defined by the program, in which case an exploit may be viewed as seeking to alter one or more of the states (or transitions) from those defined by the program.

Malware may be construed broadly as computer code that executes an exploit to take advantage of a vulnerability, for example, to harm or co-opt operation of an electronic device or misappropriate, modify or delete data. Conventionally, malware is often said to be designed with malicious intent. An object may constitute or contain malware.

The term "transmission medium" is a physical or logical communication path between two or more electronic devices (e.g., any devices with data processing and network connectivity such as, for example, a security appliance, a server, a mainframe, a computer such as a desktop or laptop, netbook, tablet, firewall, smart phone, router, switch, bridge, etc.). For instance, the communication path may include wired and/or wireless segments. Examples of wired and/or wireless segments include electrical wiring, optical fiber, cable, bus trace, or a wireless channel using infrared, radio frequency (RF), or any other wired/wireless signaling mechanism.

In certain instances, the terms “detected” and “verified” are used herein to represent that there is a prescribed level of confidence (or probability) on the presence of an exploit within an object under analysis. For instance, the IPS logic (described below) “detects” a potential exploit by examining characteristics or features of an object under analysis, and, in response, determining whether the object has characteristics indicative of an exploit (a "suspect object"). This determination may be conducted through analysis as to whether there exists at least a first probability of the object under analysis being an exploit. Likewise, the virtual execution logic "verifies" the presence of the exploit by monitoring or observing unexpected or anomalous behaviors or activities, and, in response, determining that suspect object is an exploit. According to one embodiment of the disclosure, the determination by the virtual execution logic may involve an analysis as to whether there exists a second probability of the suspect exploit being an exploit. The second probability may be greater than the first probability and may take into account the first probability.

The term "computerized" generally represents that any corresponding operations are conducted by hardware in combination with software and/or firmware. Also, the terms "compare" or "comparison" generally mean determining if a match (e.g., a certain level of correlation) is achieved between two items where one of the items may include a particular signature pattern.

Lastly, the terms "or" and "and/or" as used herein are to be interpreted as inclusive or meaning any one or any combination. Therefore, "A, B or C" or "A, B and/or C" mean "any of the following: A; B; C; A and B; A and C; B and C; A, B and C." An exception to this definition will occur only when a combination of elements, functions, steps or acts are in some way inherently mutually exclusive.

The invention may be utilized for detection, verification and/or prioritization of malicious content such as exploits. As this invention is susceptible to embodiments of many different forms, it is intended that the present disclosure is to be considered as an example of the principles of the invention and not intended to limit the invention to the specific embodiments shown and described.

## First Embodiment - IPS Logic with Virtual Execution Logic Verification

### Communication flow

Referring to FIG. 1A, an exemplary block diagram of an operational flow of threat detection and prevention within an electronic device 100 is shown. Herein, some or all of the incoming objects 110 associated with monitored network traffic are received by IPS logic 120, which is part of the static analysis engine of FIGs. 2A-2B for example. The IPS logic 120 is configured as a capture and filter device that receives the incoming objects

110 and filters, using at least exploit signatures and/or vulnerability signatures, which objects are to be provided for more in-depth analysis. The exploit signatures and/or vulnerability signatures may be updated in a periodic or aperiodic manner.

More specifically, a suspected exploit may be detected by conducting exploit signature checks and/or vulnerability signature checks, namely comparing an object under analysis to one or more pre-stored exploit signatures and/or vulnerability signatures to determine if a match is detected. In general, an "exploit signature" includes information directed to a previously detected or known attack pattern while a "vulnerability signature" includes information that characterizes a potential attempt to capitalize on a previously detected or known vulnerability, even when no specific exploit for that vulnerability is known. According to one embodiment of the disclosure, the vulnerability signature may be considered a protocol state machine that maintains state and is normally configured to define parameters for an object being a set of flows that represent an attempt being made to capitalize on a particular software vulnerability that the vulnerability signature is attempting to protect.

Upon conducting at least exploit signature checks and/or vulnerability signature checks on the incoming objects 110 and identifying a first subset of objects 130 having characteristics indicative of an exploit ("suspect objects"), the IPS logic 120 provides the first set of suspect objects 130 to verification logic 150 and provides results 140 of its analysis (referred to herein as "IPS-based results") to reporting logic 170 for storage and subsequent access.

It is contemplated that the first subset of objects 130 may be lesser in number (and potentially significantly less in number) than the incoming objects 110. For example, while the first subset of objects 130 may be a stream of objects, for ease of discussion in this section, the first subset of objects 130 may refer to at least one incoming object initially suspected of being an exploit (e.g., a suspect object matches a pre-stored exploit signature or a vulnerability signature). Hence, the IPS logic 120 routes the suspect object 130 to verification logic 150 and outputs the IPS-based results 140 associated with suspect object 130 to reporting logic 170.

The IPS-based results 140 may provide details directed to one or more suspected exploits within the suspect object 130. As an example, the details may include (i) an exploit identifier such as a particular name/family of the suspected exploit (if known); (ii) source address (e.g., Uniform Resource Locator "URL", Internet Protocol "IP" address, etc.) of the electronic device sending the suspect object; (iii) time of analysis; (iv) information associated with anticipated anomalous activities that may be conducted by the suspected exploit; (v) information regarding anticipated communication deviations from the protocol applicable to the network traffic; and/or (vi) recommended remediation techniques for this type of exploit.

As mentioned above, the suspect object 130 is routed to verification logic 150 (e.g., virtual execution logic being part of a dynamic analysis engine 270 as illustrated in FIGs. 2A-2B). The verification logic 150 attempts to verify whether the suspect object 130 is an exploit by virtual processing content within the suspect object 130 and monitoring behaviors during such virtual processing, as described below.

The results 160 of this analysis are output from the verification logic 150 for subsequent use by reporting logic 170 in generating a report 180 that visibly denotes and filters the suspect objects from the first set of objects 130 that have been verified (verified exploits) from those suspect objects from the first set of objects 130 that have not been verified (non-verified exploits). Although not illustrated in FIG. 1A, the VM-based results 160 may include (1) the suspect object; (2) time of analysis; (3) one or more scores that may be used to verify that the suspect object is likely an exploit, and if so: (i) the exploit identifier; (ii) characteristics or anomalous behaviors associated with the verified exploit, which may include video/images of anomalous behaviors; and/or (iii) name and/or version number of software detected to be vulnerable to the verified exploit.

Thereafter, at least portions of the IPS-based results 140 and the VM-based results 160 for the suspect object are combined. More specifically, in the event that the VM-based results 160 indicate that the verification logic 150 failed to verify that the suspect object 130 is an exploit (e.g., a computed score below a prescribed threshold), some or all of the IPS-based results 140 and the VM-based results 160 for that object are combined and added as part of "non-verified exploit information" 190 for storage and use by the reporting logic 170.

However, when the VM-based results 160 indicate that the verification logic 150 has verified that the suspect object 130 is an exploit (e.g., the computed score is equal to or above a prescribed threshold), some or all of the IPS-based results 140 and the VM-based results 160 may be modified to achieve a highlighted display of at least the verified exploits. For example, certain portions of the results 140 and/or 160 may be associated with display commands, which are recognized by a display controller being part of display logic within the reporting logic 170 and causes the display logic to produce an output that may visibly denotes differences between displayed results associated with verified exploits from displayed results associated with the non-verified exploits. This exploit information associated with the verified exploit may be stored as part of the "verified exploit information" 195".

The display logic 290 also may be configured to recognize that the verified exploit information 195 is to be displayed more prominently than the non-verified exploit information 190. For instance, display logic 290 may be configured to prominently display the verified exploit information within different display screens, within different display windows, within a certain section of a display screen, or positioned at a top of a listing. Additionally or in the alternative, at least a portion of the verified exploit information for each verified exploit may be conveyed using a different font (e.g., color, size, type, style, and/or effects) than the font used for conveying exploit information associated with non-verified exploits. Additionally or in the alternative, one or more images may be placed proximate to exploit information associated with each verified exploit. Illustrative examples of screen displays are shown in FIGs. 5A-5B.

Besides displaying the exploit information, the reporting logic 170 may issue an alert (e.g., by email or text message) to security administrators for example, communicating the urgency in handling one or more verified exploits. The reporting logic 170 may also issue alerts for one or more non-verified exploits by providing alerts in a manner that denotes to users a selected threat level.

As further shown, the IPS logic 120 may be communicatively coupled to a network 105 (e.g., public or private network) to receive incoming objects 110, such as one or more flows for example, destined for a particular client device. The IPS logic 120 is configured to conduct exploit signature checks and/or vulnerability signature checks on the incoming objects 110 to determine whether any of the objects 110 have characteristics indicative of an exploit, and thereafter, provide the suspect object(s) 130 to verification logic 150.

According to one embodiment of the disclosure, the communicative coupling between the IPS logic 120 and the verification logic 150 is provided in a sideband configuration, where the suspect object(s) 130 (or a copy thereof) may be temporarily stored and processed in the verification logic 150 concurrently with analysis of other objects by the IPS logic 120. This allows for the detection of exploits through a longer duration of analysis by the verification logic 150 (e.g., longer processing and monitoring of processing of the suspect object 130 within the virtual execution logic). This also allows detection of exploits with delayed activation, including time-bombs. However, it is contemplated that the IPS logic 120 may be configured in-line with verification logic 150 as shown in FIG. 1B. Herein, the IPS logic 120 may provide both the suspect objects 130 and IPS-based results 140 to the verification logic 150, where the IPS-based results may be subsequently routed to reporting logic 170 from the verification logic 150.

B. General architecture – first embodiment

Referring to FIG. 2A, an exemplary block diagram of a communication system 200 deploying a plurality of threat detection and prevention (TDP) systems $210_1$-$210_N$ (N>1, e.g., N=3) communicatively coupled to a management system 220 via a network 225 is shown. In general, management system 220 is adapted to manage TDP systems $210_1$-$210_3$. For instance, management system 220 is responsible for automatically updating one or more exploit signatures and/or vulnerability signatures used by IPS logic within some or all of TDP systems $210_1$-$210_N$. Each of these signatures may represent a prior detected exploit or an uncovered software vulnerability. Such sharing may be conducted automatically or manually uploaded by an administrator. Also, such sharing may be conducted freely among the TDP systems $210_1$-$210_3$ or subject to a subscription basis.

Herein, according to the embodiment illustrated in FIG. 2A, a first TDP system $210_1$ is an electronic device that is adapted to analyze information associated with network traffic routed over a communication network 230 between at least one server device 232 and at least one client device 234. The communication network 230 may include a public network such as the Internet, in which case an optional firewall 236 (represented by dashed lines) may be interposed prior to accessing client device 234. Alternatively, the communication network 230 may be a private network such as a wireless data telecommunication network, wide area network, a type of local area network (LAN), or a combination of networks.

As shown, the first TDP system $210_1$ may be communicatively coupled with the communication network 230 via a network interface 238. In general, the network interface 238 operates as a data capturing device (sometimes referred to as a "tap" or "network tap") that is configured to receive data propagating to/from the client device 234 and provide at least some of this data to the first TDP system $210_1$. Alternatively, as shown in FIG. 2B, the first TDP system $210_1$ may be positioned behind the firewall 236 and in-line with client device 234.

According to one embodiment of the disclosure, the network interface 238 is capable of receiving and routing objects associated with network traffic to the first TDP system $210_1$. The network interface 238 may provide the entire object or certain content within the object, for example, one or more files that are part of a set of flows, packet payloads, or the like. In some embodiments, although not shown, network interface 238 may be contained within the first TDP system $210_1$.

According to an embodiment of the disclosure, the network interface 238 may be further configured to capture metadata from network traffic intended for client device 234. According to one embodiment, the metadata may be used, at least in part, to determine protocols, application types and other information that may be used by logic within the first TDP system $210_1$ to determine particular software profile(s). The software profile(s) are used for selecting and/or configuring a run-time environment in one or more virtual machines selected or configured as part of the dynamic analysis engine 270, as described below. However, according to another embodiment, a "matched" vulnerability signature may be used for VM configuration to specify software profile(s) (or corresponding software image(s)) having the specific vulnerability associated with the matched vulnerability signature. These software profile(s) may be directed to different versions of the same software application for fetching corresponding software image(s) from storage device 265.

It is contemplated that, for any embodiments where the first TDP system $210_1$ is implemented as an dedicated appliance or a dedicated computer system, the network interface 238 may include an assembly integrated into the appliance or computer system that includes a network interface card and related logic (not shown) for connecting to the communication network 230 to non-disruptively "tap" network traffic propagating through firewall 236 and provide either a duplicate copy of at least a portion of the network traffic or at least a portion the network traffic itself to a static analysis engine 250. In other embodiments, the network interface 238 can be integrated into an intermediary device in the communication path (e.g., firewall 236, router, switch or other networked electronic device, which in some embodiments may be equipped with SPAN ports) or can be a standalone component, such as an appropriate commercially available network tap. In virtual environments, a virtual tap (vTAP) can be used to duplicate files from virtual networks.

As further shown in FIG. 2A, the first TDP system $210_1$ comprises the static analysis engine 250, a database 255, a scheduler 260, a storage device 265, a dynamic analysis engine 270, an optional classification logic 285, and a display logic 290. It is contemplated that the functionality of the classification logic 285 may be integrated into the display logic 290, where the display logic 290 would be configured with the prioritization logic 286 and/or the tag image generation logic 288.

In some embodiments, as shown in FIGs. 2A-2B, static analysis engine 250 may include one or more software modules that, when executed by one or more processors, performs multi-level static scanning on a particular object, namely exploit signature checks and/or vulnerability signature checks by IPS logic 120. Such signature check operations may involve accessing pre-stored signatures from one or more non-transitory storage mediums such as signature database 251. The static analysis engine 250 and the dynamic analysis engine 270 may be one or more software modules executed by the same processor or different processors, where these different processors may be located within the same processor package (e.g., different processor cores) and/or located at remote or even geographically remote locations that are communicatively coupled (e.g. by a dedicated communication link) or a network.

In general, referring to FIG. 2A, the static analysis engine 250 is communicatively coupled to receive one or more objects from network traffic which may be related or unrelated to each other. For instance, one object may be a series of HTTP packets operating as a flow routed over communication network 230. The static analysis engine 250 comprises IPS logic 120, where the IPS logic 120 analyzes each of the objects for known exploits using exploit signatures as well as for the protocol activity using vulnerability signatures. For instance, the exploit matching logic 252 within the IPS logic 120 performs exploit signature checks, which may involve a comparison of one or more pre-stored exploit signatures (pre-configured and predetermined attack patterns against the suspect object) from signature database 251. Similarly, the signature matching logic 253 within the IPS logic 120 performs vulnerability signature checks, which may involve a process of uncovering deviations in messaging practices set forth in applicable communication protocols (e.g., HTTP, TCP, etc.). As an illustrative example, HTTP messages may be analyzed to determine compliance with certain message formats established for the protocol (e.g., out-of-order commands). Furthermore, payload parameters of the HTTP messages may be analyzed to determine further compliance.

Upon detecting a match during the exploit signature check and/or the vulnerability signature check (an object under analysis has characteristics that suggest the object is an exploit), the IPS logic may be adapted to upload the IPS-based results 140 for storage in database 255. These results 140 may include, but are not limited or restricted to (i) an exploit identifier such as a particular name/family of the suspected exploit (if known); (ii) source address (e.g., Uniform Resource Locator “URL”, Internet Protocol “IP” address, etc.) of a source of the suspect object; (iii) time of analysis; (iv) information associated with anticipated anomalous activities that may be conducted by the suspect exploit; (v) information regarding anticipated communication deviations from the protocol applicable to the network traffic; and/or (vi) recommended remediation techniques. The IPS-based results 140 may be accessible by classification logic 285 and/or display logic 290, as described below.

Furthermore, the IPS logic 120 routes suspect object to the virtual execution logic 150 within dynamic analysis engine 270. The dynamic analysis engine 270 is configured to provide more in-depth analysis of suspect object(s) from the IPS logic 120 by analyzing the content of the suspect object(s) in order to verify whether or not the suspect object is an exploit. Additionally, according to one embodiment of the disclosure, a tag value may accompany or be associated with the suspect object for use in subsequently locating the suspect object’s corresponding stored IPS-based results 140 after virtual processing within the dynamic analysis engine 270. For instance, the tag value may be an address, an index number, or the like. It is contemplated that tag value may be separate from the suspect object or may be strategically placed within the suspect object itself (e.g., within a header portion, payload, etc.).

More specifically, after static scanning has been completed, the IPS logic 120 provides the suspect object to the dynamic analysis engine 270 for in-depth dynamic analysis using virtual machines (VMs) $275_1$-$275_M$ (M>1). For instance, the dynamic analysis engine 270 may simulate transmission and/or receipt by a destination device comprising the virtual machine. Of course, if the object is not suspected of being an exploit, the IPS logic 120 may simply store the IPS-based results within database 255 and denote that the object is benign.

According to one embodiment, one or more VMs $275_1$-$275_M$ within the virtual execution environment 272 may be configured based on the results of the exploit signature check and the vulnerability signature check conducted by the IPS logic 120. For instance, for an unknown vulnerability, the VMs $275_1$-$275_M$ may be configured with all of the software profiles corresponding to the software images stored within storage device 265. Alternatively, the VMs $275_1$-$275_M$ may be configured according to a prevalent software configuration, software configuration used by an electronic device within a particular enterprise network (e.g., client device 234), or an environment that is required for the object to be processed, including software such as a web browser application, PDF™ reader application, or the like. However, for a known vulnerability which occurs after a successful match during a vulnerability signature check, the VMs $275_1$-$275_M$ may be more narrowly configured to software profiles associated with vulnerable software.

As a first illustrative example, upon determining that the suspect object matches a particular vulnerability signature, the scheduler 260 may determine (1) what vulnerability signature has been tagged; (2) if the vulnerability is a server side vulnerability or client side vulnerability; and/or (3) which software image(s) are associated with software having the vulnerability associated with the tagged vulnerability signature. Thereafter, the software profile(s) are selected by the scheduler 260 to fetch these software image(s) for configuration of VM $275_1$. This tailored selection scheme avoids VM configuration for software that does not feature the matched (tagged) software vulnerability.

As a second illustrative example, the scheduler 260 may be adapted to configure the multiple VMs $275_1$-$275_M$ for concurrent virtual execution of a variety of different versions of the software in efforts to verify that the suspect object identified by the signature matching logic 253 is an exploit.

Of course, it is contemplated that the VM configuration described above may be handled by logic other than the scheduler 260. For instance, although not shown, the static analysis engine 250 may include configuration logic that is adapted to determine (1) what vulnerability signature was tagged; (2) if the vulnerability is a server side vulnerability or client side vulnerability; and/or (3) which software image(s) are associated with software having the vulnerability associated with the tagged vulnerability signature. This configuration logic may transmit the VM configuration information to the scheduler 260 and/or dynamic analysis engine 270 to handle VM configuration as described above.

According to one embodiment of the disclosure, the dynamic analysis engine 270 is adapted to execute one or more VMs $275_1$-$275_M$ to simulate the receipt and execution of content associated with an object under analysis within a run-time environment as expected by the type of object. For instance, dynamic analysis engine 270 may optionally include a protocol sequence replayer (replay logic) 280 to replay the suspect object and provide replayed data flows to the VM(s) $275_1$,…, and/or $275_M$ or object extractor logic 282 to extract a self-contained object within a data flow for virtual processing by VM(s) $275_1$,…, and/or $275_M$. One embodiment of the protocol sequence replayer is described in U.S. Patent No. 8,375,444, the entire contents of which are incorporated by reference herein.

For example, the replay logic 280 may be adapted to provide, and sometimes modify (e.g. modify IP address, etc.) packets associated with the suspect objects and synchronize any return network traffic generated by the virtual execution environment 272 in response to the packets. Hence, the replay logic 280 may suppress (e.g., discard) the return network traffic such that the return network traffic is not transmitted to the communication network 230. According to one embodiment of the disclosure, for a particular suspect object being a flow such as a TCP or UDP sequence, the replay logic 280 may replay the data packets by sending packets to the virtual execution environment 272 via a TCP connection or UDP session. Furthermore, the protocol sequence replay logic 280 synchronizes return network traffic by terminating the TCP connection or UDP session.

As further shown in FIG. 2A, the monitoring logic 276 within the dynamic analysis engine 270 may be configured to monitor behavior of the content being analyzed by one or more VMs $275_1$,…, and/or $275_M$, for detecting anomalous or unexpected activity indicative of an exploit. If so, the content may be determined as being associated with malicious activity, and thereafter, monitoring logic 276 operating with a score determination logic 278 may route the VM-based results 160 (e.g., computed score, information associated with the detected anomalous behaviors, and other information associated with the detected malicious activity by the suspect object) to classification logic

285 and/or database 255. It is noted that the tag value, if used, may be provided as part of the VM-based results 160.

According to one embodiment of the disclosure, the score determination logic 278 comprises one or more software modules that are used to determine a probability (or level of confidence) that the suspect object is an exploit. Score determination logic 278 is configured to generate a value (referred to as a "score") that classifies the threat of the possible exploit. Of course, a score may be assigned to the suspect object as a whole by mathematically combining the scores determined by analysis of different content associated with the same suspect object to obtain an overall score for that suspect object. Thereafter, the suspect object and/or score are routed to classification logic 285 for use in prioritization.

In general, the classification logic 285 may be configured to receive the VM-based results 160. According to one embodiment of the disclosure, the score may be used, at least in part, to determine whether the virtual execution logic 150 has verified that the suspect object is an exploit. Where the score represents that the suspect object 130 has not been verified by the virtual execution logic 150 to have the characteristics of an exploit, some or all of the VM-based results 160 may be combined with its corresponding IPS-based results to produce the non-verified exploit information 190, which is stored in database 255.

However, if the score represents that the suspect object 130 has been verified by the virtual execution logic 150 as an exploit, at least some of the combined IPS-based results 140 and/or the VM-based results 160 may be modified by the classification logic 285 and subsequently stored as at least part of the verified exploit information 195. Stated differently, the classification logic 285 operating with the database 255 may be responsible for prioritizing the display of exploit information associated with the verified exploits. This may involve the classification logic 285 modifying order or position for the displayed verified exploit information, or adding information to the verified exploit information that is used by the display logic 290 to modify display order or positioning; modifying the type of font (e.g., color, size, type, style, and/or effects) used for text conveying certain verified exploit information; placing one or more images proximate to verified exploit information for each verified exploit; or the like.

Of course, it is contemplated that other parameters, combined with or separate from the score, may be used or relied upon to determine whether and/or how to highlight display of the exploit information associated with the suspect object.

Thereafter, along with non-verified exploit information 190, the verified exploit information 195 is stored within database 255 and accessible by display logic 290.

More specifically, according to one embodiment of the disclosure, classification logic 285 comprises prioritization logic 286 and tag image generation logic 288. According to one embodiment of the disclosure, the prioritization logic 286 may be adapted to modify (e.g., alter or associate display commands to) exploit information associated with verified exploits based one or more factors, including (i) score associated with the object; (ii) source of the object; (iii) repeated detection of the same exploit in different suspect objects; or the like. This modification may involve modifying font (e.g., color, size, type, style, and/or effects) used to convey the exploit information associated with verified exploits. As another example, this modification may involve classification and storage of the exploit information as verified exploit information 195 which, when accessed by the display logic 290, places the exploit information associated with the verified exploit at a specific location on a display screen or within display image (e.g., within a specific window or display screen listing the verified exploits, at a particular order within the listing of the verified and non-verified exploits, etc.).

Of course, as an alternative, the display logic 290 may be implemented with some or all of the functionality associated with the prioritization logic 286 and/or tag image generation logic 288 in lieu of deployment within the classification logic 285. Hence, responsive to information received from the classification logic, the display logic 290 may be adapted to modify exploit information associated with verified exploits.

The tag image generation logic 288 may be adapted to operate in combination with the prioritization logic 286 to generate a tag image (not shown), which is included as part of the verified exploit information 195 associated with suspect object for display. The tag image is used to provide another visual representation of the presence of a verified exploit, namely a suspected exploit detected by the IPS logic 120 whose presence has been verified by the virtual execution logic 150.

Of course, in lieu of or in addition to static scanning operations being conducted by TDP systems $210_1$-$210_3$, it is contemplated that cloud computing services 240 may be implemented with IPS logic 120 to perform the exploit and/or vulnerability signature checks and/or with virtual execution logic 150 to conduct virtual execution on content within the object under analysis, as described herein. The display logic 290 may cause the display of the exploit information associated with the verified exploits and/or non-verified exploits graphically or otherwise through a downloaded page or pages from the cloud computing services 240 to a client device or to an application running on a client device that displays the results obtained from the cloud computing services 240. In accordance with this embodiment, TDP system $210_1$ may be adapted to establish secured communications with cloud computing services 240 for exchanging information.

C. General architecture – second embodiment

Referring now to FIG. 2B, first TDP system $210_1$ may be coupled with the communication network 230 in line with client device 234. Contrary to the embodiment illustrated in FIG. 2A, first TDP system $210_1$ comprises an interface unit 295 that directs signaling on communication network 230 to static analysis engine 250 or classification logic 285, given that the dynamic analysis engine 270 is deployed in cloud computing services 240. Hence, objects from network traffic for static analysis are routed to static analysis engine 250 via communication path 296. The suspicious objects may be routed via path 297 to the dynamic analysis engine 270 in cloud computing services 240. Similarly, objects that are not determined to be at least "suspect" may be returned via path 297 for continued routing to client device 234. The results of the dynamic analysis engine 270 (e.g., exploit information) may be routed via path 298 for prioritization and tagging before storage within database 255 for subsequent use by display logic 290.

D. Exemplary Logic Layout of TDP System

Referring now to FIG. 3, an exemplary block diagram of logic associated with TDP system $210_1$ of FIGs. 2A-2B is shown. TDP system $210_1$ comprises one or more processors 300 that are coupled to communication interface logic 310 via a first transmission medium 320. Communication interface logic 310 enables communications with other TDP systems $210_2$-$210_3$ and management system 220 of FIG. 2A-2B. According to one embodiment of the disclosure, communication interface logic 310 may be implemented as a physical interface including one or more ports for wired connectors.

Additionally, or in the alternative, communication interface logic 310 may be implemented with one or more radio units for supporting wireless communications with other electronic devices.

Processor(s) 300 is further coupled to persistent storage 330 via transmission medium 325. According to one embodiment of the disclosure, persistent storage 330 may include (i) static analysis engine 250, including first analysis logic (e.g., IPS logic) 250; (ii) the dynamic analysis engine 270, including virtual execution logic 272, monitoring logic 276, score determination logic 278 along with optional replay and object extractor logic 280 and 282; (iii) classification logic 285 including prioritization logic 286 and tag image generation logic 288; and (iv) display logic 290. Of course, when implemented as hardware, one or more of these logic units could be implemented separately from each other.

IPS logic 120 comprises one or more software modules that conduct a first static analysis on each incoming object. As described above, this analysis may involve performing at least exploit signature checks and vulnerability signature checks on each incoming object to determine whether characteristics of any of these objects are indicative of an exploit. If not, the analysis may be discontinued for the object, or the object may be provided for non-real time forensic review. Upon confirming that one or more suspect objects have characteristics of an exploit, the IPS logic 120 provides the suspect object(s) to the virtual execution logic 150. It is contemplated that a tag value, if used, may accompany (or be associated with) the suspect object to identify a stored location of the IPS-based results 140 for the suspect object, as described above. The IPS-based results 140 are uploaded to data store 350, at least partially operating as a database, for subsequent access by classification logic 285.

Virtual execution environment 272 comprises one or more software modules that are used for performing an in-depth, dynamic and real-time analysis of the suspect object using one or more VMs. More specifically, the virtual execution environment 272, protocol sequence replay logic 280 and/or object extractor logic 282 are adapted to run the VM(s), which virtually processes the content associated with the suspect objects by simulating receipt and execution of such content in order to determine the presence of one or more exploits. Furthermore, the monitoring logic 276 monitors in real-time and may also log at least anomalous behaviors by the VM(s) configured with certain software and features that are presumably targeted by the matched exploit or vulnerability. In essence, the monitoring logic 276 identifies the effects that the suspect object would have had on a physical electronic device with the same software/feature configuration. Such effects may include unusual network transmissions, unusual changes in performance, and the like.

Thereafter, according to the observed behavior of the virtually executed content, the monitoring logic 276 may determine that the content is associated with one or more exploits, where the severity of the observed anomalous behavior and/or the likelihood of the anomalous behavior results from an exploit, is evaluated and reflected in a "score" assigned by the score determination logic 278. As a result, these logic units collectively output the VM-based results 160 for use by classification logic 285 to highlight exploit information associated with verified exploits.

The prioritization logic 286 comprises one or more software modules that are used to highlight information associated with verified exploits, namely the verified exploit information 195. For instance, the prioritization logic 286 may assign higher priority to exploit information directed to verified exploits, where the priority may be used by the display logic 290 to determine an order or location for display. Furthermore, the prioritization logic 286 may be adapted to modify the font used in display of the verified exploit information (e.g., color, size, type, style, and/or effects), or control the placement of one or more images provided by the tag image generation logic 288 proximate to its corresponding exploit information.

Continuing the above example, processor(s) 300 may invoke display logic 290, which produces one or more screen displays for conveying a detailed summary of verified and/or non-verified exploits detected by the TDP system $210_1$. According to one embodiment of the disclosure, the information associated with the verified exploits (verified exploit information 195) may be presented in a first area of a display screen while information associated with the non-verified exploits (non-verified exploit information 190) may be presented in a second area of the display screen. As another example, the verified exploit information 195 may be presented as top entries in a listing of all exploits detected by the IPS logic while the non-verified exploit information 190 is presented subsequently. As another example, some or all of the verified exploit information 195 may be presented in different font (e.g., different type, color, style such as bold or italic, effects such as underlining or shadow, etc.) than font used for conveying the non-verified exploit information 190. As yet another example, a tag image may be positioned next to the verified exploit information 195 unlike non-verified exploit information 190 associated with suspect objects.

## E. Display and Prioritization of Detected Exploits

Referring to FIG. 4, an exemplary diagram of a flowchart illustrating a threat detection and prevention process which generates a report that highlights information associated with suspected exploits detected by the IPS logic and verified by the virtual execution environment is shown. Upon receipt of an object, the TDP system conducts a first static analysis operation on the object (block 400). Herein, the first static analysis operation may include exploit signature checks and/or vulnerability signature checks by the IPS logic to determine whether characteristics of an object under analysis are indicative of an exploit. Upon determining that the suspect object may have the characteristics of one or more suspected exploits, the object is tagged for VM-based analysis and information associated with the suspect object and/or potential exploit (IPS-based results) is stored for subsequent access (blocks 405 and 410).

Although not shown, when determining that the suspect object has characteristics of a suspected exploit, the IPS logic may be configured to block the object from proceeding to the targeted client device, although blocking may be delayed until completion of the VM-based analysis to mitigates errors due to false positives. This blocking functionality may be adjusted by the network administrator based on the severity/type of suspected exploit, number of occurrences of this type of exploit within a prescribed time period, or the like. Furthermore, prior to performing further exploit analysis, if used, a tag value may accompany (or being associated with) the suspect object when output from the IPS logic so that the IPS-based results for the suspect object can be related to the subsequent VM-based results for that object.

After IPS-based analysis for the suspect object has concluded, the content of the suspect object may undergo VM-based analysis (blocks 415 and 420). The results of the VM-based analysis (VM-based results) are provided for subsequent review (block 425). According to one embodiment of the disclosure, the classification logic performs such review, although in the alternative, logic within the dynamic analysis engine may conduct this review.

Normally, if the VM-based analysis fails to verify that the suspect object is an exploit, a score may be assigned to denote that no exploit has been detected (block 430). In this case, information produced during the VM analysis of the suspect object along with its corresponding IPS-based results are stored as part of the non-verified exploit information (block 435). However, during virtual execution of the object, if the monitored behavior denotes that the suspect object is an exploit, a score is assigned that represents the likelihood and/or threat level for the "verified" exploit(s).

According to one embodiment of the disclosure, the classification logic may be configured to obtain the IPS-based results associated with the verified exploit, where some or all of the information from the IPS-based results and the VM-based results may be prominently displayed (highlighted) as illustrated in blocks 440 and 445. Such highlighting may include (i) assigning a specific display location for exploit information associated with verified exploits that is different from exploit information associated with non-verified exploits; (ii) modifying the presentation (e.g., font type, color, style, etc.) of exploit information associated with verified exploits where the exploit information associated with the non-verified exploits will have a different presentation; (iii) controlling placement of one or more images proximate to exploit information associated with verified suspect objects only. Other display adjustments may be used, as this highlighting is conducted to visibly differentiate exploit information associated with the verified exploits from exploit information associated with the non-verified exploits.

Thereafter, the (highlighted) verified exploit information is uploaded into the database for storage and now accessible by display logic for rendering (blocks 450 and 455).

F. Display Screens of detected malicious objects

Referring now to FIG. 5A, an exemplary embodiment of a first user interface display screen 500 produced by the display logic of FIGs. 2A-2B that provides an interactive dashboard is shown. Herein, rendered by the display logic, the display screen 500 comprises a plurality of display areas 510 and 530 that illustrate information directed to exploits uncovered over a selected time period by the TDP system. It is noted that multiple highlighting techniques are shown in display screens 500 and 545, although it is contemplated that any one or more highlighting technique may be conducted for a particular display.

More specifically, according to one embodiment of the disclosure, a first area 510 displays a plurality of entries 5201-520R (R>1, R=6 for this embodiment) that provide information directed verified exploits and/or non-verified exploits. As shown, each row of entries (e.g., 5201) rendered by the display logic comprises a plurality of fields, including one or more of the following: (1) a name 521 of the exploit associated with a suspect object; (2) a signature pattern 522 applicable to the object under analysis; (3) addressing information 523 (e.g., Internet Protocol "IP" address, Media Access Control "MAC" address, etc.) for a source device providing the verified or non-verified exploit; (4) a level of severity 524 (e.g., high, medium, low) of the detected exploit, where the severity level corresponds, at least in part, to the threat score; (5) a time 525 during which the exploit analysis process was conducted; and/or (6) name and/or version number 526 of software detected to be vulnerable to the detected exploit.

A second area 530 may be configured with one or more images corresponding to each entry for a verified exploit, namely an object initially identified by the IPS logic as having characteristics indicative of an exploit and verified of being an exploit by the virtual execution logic. For instance, as illustrated in FIG. 5A, image 535 is displayed proximate to information associated with a corresponding verified exploit named "HTTP Exploit_ID1." Similar images are illustrated for verified exploit information associated with verified exploits named "HTTP Exploit_ID2," "Java Exploit_ID1," and "HTML Exploit_ID1."

It is noted that the mere existence of a verified exploit may warrant heightened severity level, but does not require heightened severity levels as illustrated by the fact that certain non-verified exploits may be assigned higher severity levels than some verified exploits. Rather, exploit information associated with the verified exploits is highlighted, namely this exploit information is displayed more prominently than exploit information associated with non-verified exploits for example. This allows a network administrator to more quickly and easily determine verified exploits and thereby substantially mitigate administrative and operational disadvantages from false-positives.

As an example, as a highlighting technique, the font associated with the exploit names (HTTP Exploit_ID1; HTTP Exploit_ID2; Java Exploit_ID1; and HTML Exploit_ID1) may be displayed differently than the font associated with the exploit names for non-verified exploits (e.g., Java Exploit_ID2). Alternatively, the verified exploit information associated with the verified exploits may be ordered at the top of the listing (see FIG. 5B). Also, a single display screen may produce two areas, where a first area includes exploit information associated with verified exploits while a second area includes exploit information associated with non-verified exploits (see FIG. 5B).

Furthermore, although not shown, it is contemplated that selection of a portion of the entry (e.g., entries within fields 521/522/523/524/526 (as represented by an underlined portion) and/or a separate "Details" field 540) may enable the network administrator to obtain more detailed information of the exploit and/or analysis associated with that exploit.

For instance, by selecting the particular listed exploit 521, the administrator may be able to uncover family and other information related to the exploit (e.g., documented attacks, recommended remediation techniques, targeted client device(s), etc.). Also, by selecting the signature 522, the administrator may have access to additional information concerning what signature (exploit, vulnerability, etc.) was determined by the IPS to match the suspect object. Additional information (e.g., information on signature updates, detection history of this signature with other objects, etc.) may be provided as well.

Similarly, by selecting the corresponding host address 523 or the severity level 524, the administrator may be provided with additional information directed to geographic location of the source of the suspect object corresponding to that exploit, addressing information directed to intermediary devices that received the suspect object, the particular network operations targeted by the exploit, or the like. Also, by selecting the software type 526, a listing of all software types detected to be vulnerable to the verified exploit (along with video/images of monitored anomalous behaviors denoting the presence of such exploit) may be accessed.

Referring now to FIG. 5B, an exemplary embodiment of a second user interface display screen 545 produced by the display logic of FIGs. 2A-2B that provides an interactive dashboard is shown. Herein, the display screen 545 comprises a plurality of areas 550, 570 and 580 that display results of IPS detection analysis over a selected time period.

As shown, similar to the first user interface display screen 500, first area 550 of the second user interface display screen 545 displays a plurality of entries 5601-560S (S>1, S=4 for this embodiment) that provides information directed to verified exploits. Each of the entries (e.g., 5601) rendered by the display logic comprises: (1) a name 561 of the verified exploit (suspect object verified to be an exploit); (2) a signature 562 that initially identified the suspect object as having characteristics indicative of an exploit; (3) addressing information 563 (e.g., Internet Protocol “IP” address, Media Access Control “MAC” address, etc.) for a source device providing the detected exploit; (4) a level of severity 564 (e.g., high, medium, low) of the detected exploit that corresponds, at least in part, to the threat score; (5) a time 565 during which the exploit analysis process was conducted; and/or (6) name and/or version number 566 of software detected to be vulnerable to the detected exploit.

As shown, a second area 570 may be provided, which comprises an image corresponding to each entry that is associated with the verified exploits, as described above. As illustrated in FIG. 5B, image 535 is displayed with information associated with a corresponding verified exploit named “HTTP Exploit_ID1.” Similar images are illustrated as highlighted verified exploit information for verified exploits named “HTTP Exploit_ID2,” “Java Exploit_ID1,” and “HTML Exploit_ID1.”

A third area 580 illustrates exploit information associated with non-verified exploits named “Java Exploit_ID2”, “RPC Exploit_ID1” for example.

II. Alternative embodiment - IPS Logic & Secondary Analysis logic with Virtual Execution Logic Verification

According to an alternative embodiment of the disclosure, the static analysis engine may be configured with a first static analysis logic (e.g., IPS logic) and a second static analysis logic (e.g., heuristic logic), which is configured to operate independently from the IPS logic and identifies whether characteristics of any of the incoming objects are indicative of an exploit. As described below, the first static analysis logic and the second static analysis logic may operate in parallel or in tandem.

In particular, as described above, the first static analysis logic (IPS logic) conducts at least exploit signature checks and/or vulnerability signature checks on the incoming objects to identify a first subset of objects having characteristics indicative of an exploit. The second static analysis logic (heuristic logic) may be configured to analyze the same or different objects, where such analysis may be in accordance with at least a set of rules and/or signatures different than those utilized by the first static analysis logic (IPS logic).

More specifically, according to this embodiment of the invention, upon identifying the suspect objects (first subset of objects), the first static analysis logic (IPS logic) provides suspect objects, perhaps each accompanied by or associated with a tag identifier (hereinafter referred to as "tag_ID1"), to the verification logic 150 of FIGs. 6A-6B. Tag_ID1 may be used to indicate to other logic that the suspect object originated from the first static analysis logic (IPS logic).

The second static analysis logic (heuristic logic) is configured to analyze the incoming objects to determine whether the presence, absence or modification of information within an object may denote potential malicious activity indicating that object may be an exploit. Such determination may involve the second static analysis logic (heuristic logic) conducting operations to determine whether certain portions of the object corresponds to one or more "malicious identifiers," which may include, but are not limited or restricted to a particular source or destination address (e.g., URLs, IP addresses, MAC addresses, etc.) that is associated with known exploits; exploit patterns; or shell code patterns.

Additionally, with each suspect object, the heuristic logic may provide a tag identifier (tag_ID2) for use in locating corresponding heuristic-based results 640 associated with each suspect object 630. Hence, tag_ID2 may be further used to identify to other logic that this suspect object originated from the heuristic logic 620.

After either the first static analysis logic (IPS logic) or the second static analysis logic determine which of the incoming objects have characteristics indicative of an exploit, the suspect objects are provided to the virtual execution logic for more in-depth dynamic analysis using one or more virtual machines (VMs). Such dynamic analysis may include virtual execution of the content of the suspect objects with one or more configured VMs, as described above. The behaviors of the VM(s) are monitored for detection of anomalous or unexpected activity.

It is contemplated that the first static analysis logic (IPS logic) and the second static analysis logic (heuristic logic) may operate in parallel in which both of these logic units conduct the preliminary exploit detection analysis on the same suspect objects. More specifically, the second static analysis logic (heuristic logic) may conduct its analysis on an object extracted from the network traffic concurrently (i.e. at least partially overlapping in time) with the analysis of the same object by the IPS logic. This provides the TDP system with an ability to account for false negatives that signify a lack of detection of an exploit by the IPS logic. Also, such parallel analysis may be conducted in order to increase scrutiny of network traffic for objects originating from a certain geographic location prone to exploits, from a certain IP addresses that have been identified as a malicious source, or the like.

Of course, it is contemplated that the first static analysis logic (IPS logic) and second static analysis logic (heuristic logic) may operate in tandem in which an incoming object is capable of being processed by either the IPS logic or the heuristic logic within the embodiment. Control of the selection as to whether the static analysis is performed by the first static analysis logic (IPS logic) or the second static analysis logic (heuristic logic) may be assigned to additional control logic within the static analysis engine. Such control may be based on the type of object under analysis, source, traffic conditions, or the like.

A. General Communication Flow

Referring to FIG. 6A, an exemplary block diagram of an operational flow of threat detection and prevention within an electronic device 600 is shown. Herein, some or all of the incoming objects 110 associated with the monitored network traffic may be received by a first static analysis logic (e.g., IPS logic 120 of FIG. 1A), as described above. The IPS logic 120 is configured to perform at least exploit signature checks and/or vulnerability signature checks on some or all of the incoming objects 110.

Upon identifying that a first subset 610 of the incoming objects 110 are "suspicious" (e.g., one or more objects 110 match an exploit signature and/or vulnerability signature), the IPS logic 120 subsequently routes the first subset of suspect objects 610 to the verification logic 150 (e.g., virtual execution logic). Each of these objects may be accompanied by a tag identifier (tag_ID1) and provided to the verification logic 150.

Besides being used for subsequently locating the IPS-based results 140 associated with the suspect object (provided from the IPS logic 120 to the reporting logic 170), tag_ID1 may be used to additionally to identify to the verification logic 150 and/or reporting logic 170 that these suspect objects 610 are provided from the IPS logic 120.

Such information may be useful for identifying exploit information associated with verified exploits originating from the IPS logic, where this exploit information may be highlighted even differently than exploit information associated with verified exploits originating from a second static analysis logic 620.

Operating in tandem or in parallel with IPS logic 120, the second static analysis logic 620 (e.g., heuristic logic) conducts another type of static analysis on some or all of the objects 110 to produce a subset of objects 630 having characteristics indicative of an exploit. Hence, when operating in parallel, heuristic logic 620 may receive the incoming objects 110, which are also being received and analyzed by IPS logic 120. When operating in tandem with the IPS logic 120, the heuristic logic 620 may receive some or all of the incoming objects 110, where the switching between receipt of specific incoming objects by either the IPS logic 120 or the heuristic logic 620 may be conducted by switching logic 645 via control signals 647 from scheduler 260 or some other logic within TDP system $210_1$, as shown in FIG. 6B.

The suspect objects 610 and/or 630 (collectively referred to as “suspect objects 635”), detected by the IPS logic 120 and/or heuristic logic 620, are routed to the verification logic 150. The verification logic 150 is adapted to verify whether any of the suspect objects is an exploit through virtual processing of the content within these objects 635. The VM-based results 650 of this analysis are output from the verification logic 150 for subsequent use by reporting logic 170 for display purposes, as described above.

More specifically, the first static analysis logic (e.g., IPS logic 120) conducts at least exploit signature checks and/or vulnerability signature checks to identify whether characteristics of any of the analyzed objects 110 are indicative of an exploit. If so, the IPS logic 120 forwards these suspect object(s) 610 to the verification logic 150.

Additionally, one or more heuristic checks may be conducted on some or all of objects 110, including various scanning operations conducted on portions of the objects to determine correspondence with one or more malicious identifiers, as described above. While the IPS logic 120 is adapted to identify objects in accordance with at least exploit signature checks and/or vulnerability signature checks, the heuristic checks are directed to a more expansive static analysis of some or all of objects 110, including the use of different types of signatures or other static analysis schemes.

After performing the heuristic check(s) by the heuristic logic 620, a second set of suspect objects 630 is provided to the verification logic 150. Again, the second set of objects 630 may be lesser (and potentially significantly less) in number than the incoming objects 110.

After virtual processing of content within each of the suspect objects 610 and/or 630, and thereafter verifying that particular objects are exploits (verified exploits), the verification logic 150 provides VM-based results 650 that may be modified, along with its corresponding IPS-based results 140, to generate a report 660 (e.g., one or more display screens, printed report, etc.). The report 660 is configured to visibly highlight exploit information associated with verified exploits. As an alternative, the report 660 may also be configured to visibly highlight exploit information associated with verified exploits from exploit information associated with non-verified exploits (suspect objects having characteristics of exploits that were not verified by the VMs).

B. General architecture

Referring to FIG. 7A, an exemplary block diagram of a communication system 700 deploying a plurality of threat detection and prevention (TDP) systems 7101-710N (N>1, e.g., N=3) is shown. TDP system 7101 is identical to TDP system $210_1$ of FIG. 2A, except that static analysis engine 750 includes two different static analysis logic units. More specifically, as shown in FIGs. 7A and 7B, static analysis engine 750 may include one or more software modules that, when executed by one or more processors, performs multi-level static scanning on a particular object, namely both exploit and vulnerability signature checks by IPS logic 120 and heuristic checks by heuristic logic 620.

Operating in parallel or tandem with IPS logic 120, the heuristic logic 620 is configured to conduct one or more heuristic checks on objects under analysis. These heuristic checks may be considered more expansive in analysis than the exploit and/or vulnerability checks conducted by the IPS logic 120 as mentioned above.

Herein, based on the results of the heuristic checks conducted by heuristic logic 620, score determination logic 720 determines the probability (or level of confidence) that the characteristics of the analyzed object are indicative of an exploit. In other words, score determination logic 720 is configured to generate a value that classifies the threat level of the possible exploit characterized by each of the analyzed objects. For instance, if the heuristic checks detect one type of characteristic that suggests the object under analysis is an exploit, the object may be classified with a first threat level. The first threat level may be represented by a score (value) corresponding to the likelihood of the object being an exploit (e.g., score of 3 out of 10). However, if the heuristic checks detect multiple characteristics or another type of characteristic that more strongly suggests the object under analysis is an exploit, a higher score (e.g., score of 8 out of 10) may be assigned by score determination logic 720 to denote a higher probability of the detected presence of an exploit.

Thereafter, the objects and their corresponding scores may be routed from the static analysis engine 750 to the dynamic analysis engine 270 for use in further analysis to verify which of the suspect objects, if any, are exploits. Additionally or in the alternative, it is contemplated that the score may be provided to classification logic 785 for use in prioritization.

More specifically, after static scanning has completed, the object may be provided to the dynamic analysis engine 270 for in-depth dynamic analysis using virtual machines (VMs) $275_1$-$275_M$ (M>1). Of course, if the characteristics of the object are not indicative of an exploit, the heuristic logic 620 may halt further analysis of content with the object.

In general, besides receiving VM-based results 160 from dynamic analysis engine 270, the classification logic 785 may be configured to receive assigned scores from static analysis engine 750. Classification logic 785 may be configured to mathematically combine the scores assigned to content associated with the suspect object (based on findings from static analysis engine 750 and dynamic analysis 270) to obtain an overall score that is assigned with the verified or non-verified exploit.

According to one embodiment of the disclosure, the overall score may be used, at least in part, to identify verified exploits from non-verified exploits. Also, the score may be used, at least in part, for highlighting operations such as assigning a display priority that may influence the display ordering as described above. However, it is contemplated that other parameters, combined with or separate from the score assigned to the exploit, may be used to classify exploits or influence display priority. For instance, the overall score along with other parameters, such as the presence of the tag_ID1 or tag_ID2 as part of exploit information included in the VM-based results, may influence the display ordering of that exploit.

Referring now to FIG. 7B, first TDP system 7101 may be coupled with the communication network 230 in line with client device 234. As similarly illustrated in FIG. 2B, first TDP system 7101 comprises an interface unit 295 that directs signaling on communication network 230 to static analysis engine 750 or classification logic 785, given that the dynamic analysis engine 270 is deployed in cloud computing services 240.

### C. Display and Prioritization of Detected Exploits

Referring to FIGs. 8A-8B, an exemplary diagram of a flowchart illustrating a threat detection and prevention process, utilizing IPS logic and/or heuristic logic for static analysis, is shown, where the process generates a report that highlights information associated with suspected exploits detected by the IPS or heuristic logic and verified by the virtual execution environment. Herein, the IPS logic and the heuristic logic may operate in parallel or in tandem.

The IPS logic and heuristic logic may be configured to operate in parallel (or in tandem) based on factors that may warrant increased scrutiny in efforts to detect exploits. For instance, there is an increased amount of objects originating from a certain geographic location prone to exploits or from a certain IP address that has been identified as a malicious source. For parallel processing, operations associated with blocks 805-825 and 830-855 of FIG. 8A are conducted in parallel. For this discussion, however, the IPS logic and heuristic logic are operating in tandem. Also, for certain governmental agencies, its sensitivity to exploits and/or its history in experiencing exploits may warrant additional analysis.

Upon receipt of an object under analysis, as set forth in block 800, the TDP system conducts a determination as to whether the static analysis should be conducted by the first static analysis logic (IPS logic) and/or the second static analysis logic (heuristic logic). According to one embodiment, as a default, the IPS logic is selected.

When selected, the IPS logic conducts exploit signature checks and/or vulnerability signature checks to determine whether characteristics of the object under analysis are indicative of an exploit (block 805). Upon determining that the characteristics of the object under analysis are indicative of an exploit, information associated with the suspect object and/or exploit (IPS-based results) is stored for subsequent access (blocks 810 and 815).

Although not shown, when determining that the suspect object has characteristics of a suspected exploit, the IPS logic may be configured to block the object from proceeding to the targeted client device, although blocking may be delayed until completion of the VM-based analysis. This blocking functionality may be adjusted by the network administrator based on the severity/type of suspected exploit, number of occurrences of this type of exploit within a prescribed time period, or the like. Furthermore, prior to performing further exploit analysis, as an optional feature identified by dashed lines in FIG. 8A, tag_ID1 may accompany the suspect object when output from the IPS logic so that (1) the IPS-based results for the suspect object can be related to the subsequent VM-based results for that object and (2) the virtual execution logic and/or classification logic can identify that the suspect object originated from the IPS logic (block 820). Thereafter, the suspect object and/or tag_ID1 is provided to the dynamic analysis engine for subsequent analysis (block 825).

Additionally or in the alternative, a second static analysis may be performed to determine whether characteristics of the object under analysis are indicative of an exploit (block 830). This determination may involve one or more heuristic checks being conducted in efforts to determine if the (i) the object has a certain level of correlation with one or more malicious identifiers or (ii) presence, absence or modification of any content associated with the object identifies a potential exploit. During such analysis, a score may be assigned to identify the likelihood of this object being an exploit (block 835).

In the event that the suspect object is tagged for VM-based analysis, which may be determined if the assigned score is greater than or equal to a prescribed threshold score, information associated with the suspect object and/or the potential exploit including the score (hereinafter referred to as “heuristic-based results”) may be stored for subsequent access by classification logic (blocks 840 and 845). Thereafter, the suspect object, optionally with tag_ID2, is provided to the dynamic analysis engine for subsequent analysis (blocks 850 and 855).

Regardless whether the static analysis is conducted by the IPS logic or the heuristic logic, the suspect object may be further analyzed by conducting VM-based analysis on the content associated with the suspect object, where behaviors of the virtual processing of the content by one or more VMs produces VM-based results (blocks 860 and 865). If the VM-based analysis fails to detect any exploit within content of the suspect object, a score may be assigned to denote that no exploit is detected and the VM-based results may be stored (blocks 870 and 875).

However, when the dynamic analysis engine verifies (during virtual processing of the content within the suspect object) that the suspect object constitutes an exploit, this “verified” exploit is assigned a score representative of the likelihood and/or threat level for the detected exploit(s). More specifically, during subsequent analysis of the content within the suspect object by the virtual execution logic, upon determining that the suspect object is an exploit (e.g., a certain probability that content within the suspect object constitutes an exploit is determined), a score representative of the likelihood and/or threat level for the detected exploit is assigned.

Thereafter, according to one embodiment of the disclosure, the IPS-based results along with the VM-based results are obtained and some or all of the information from the IPS-based results and the VM-based results may be prominently displayed (highlighted) as illustrated in blocks 880 and 885 and further described above.

Thereafter, the (highlighted) verified exploit information is uploaded into the database for storage and now accessible by display logic for rendering (blocks 890 and 895).

In the foregoing description, the invention is described with reference to specific exemplary embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention as set forth in the appended claims.

## CLAIMS

What is claimed is:

1. A threat detection system, comprising:

an intrusion protection system (IPS) logic configured to receive a first plurality of objects and analyze the first plurality of objects in accordance with a signature check to identifying a second plurality of objects as suspicious objects, the second plurality of objects being a subset of the first plurality of objects and being lesser or equal in number to the first plurality of objects; and

a virtual execution logic configured to receive the second plurality of objects and verify whether any of the second plurality of objects is an exploit, the virtual execution logic including at least one virtual machine configured to virtually process content within each of the second plurality of objects and monitor for anomalous behaviors during the virtual processing that are indicative of exploits.

2. The threat detection system of claim 1, wherein the virtual execution logic verifies whether any of the second plurality of objects is an exploit by (i) analyzing incoming suspicious objects that at least partially include the second plurality of objects received from the IPS logic, and (ii) determining a first subset of suspicious objects associated with the second plurality of objects as being exploits when the anomalous behaviors monitored during virtual processing of the first subset of suspicious objects are indicative of exploits, the first subset of suspicious objects being lesser in number than the second plurality of objects.

3. The threat detection system of claim 1 or 2 further comprising display generation logic that generates a display of exploit information for exploits as determined by the virtual execution logic, the display generation logic visually representing a display of exploit information associated with the first subset of suspicious objects associated with the second plurality of objects differently than a display of exploit information associated with a second subset of the suspicious objects that are received by logic other than the IPS logic.

4. The threat detection system of claim 1 or 3, wherein the IPS logic analyzes the first plurality of objects in accordance with the signature check operating as either (1) an exploit signature check or (2) a vulnerability signature check,

wherein the exploit signature check compares each of the first plurality of objects to one or more exploit signatures, wherein an object of the first plurality of objects is identified as a suspicious object upon matching an exploit signature of the one or more exploit signatures, and

wherein the vulnerability signature check compares each of the first plurality of objects to one or more vulnerability signatures, wherein an object of the first plurality of objects is identified as a suspicious object upon matching a vulnerability signature that characterizes a sequence of communications that indicate an attempt to attack a software vulnerability protected by the vulnerability signature.

5. The threat detection system of claim 3, wherein the exploit information comprises one or more of the following: (1) a name of an exploit; (2) a signature pattern used in detection of the exploit; (3) addressing information for a source device that provided the exploit; (4) a level of severity of the exploit, where the severity level corresponds, at least in part, to a threat score; (5) a time during which an exploit analysis process was conducted; and (6) a name or a version number of software detected to be vulnerable to the exploit.

6. The threat detection system of claim 3 or 5, wherein the display generation logic visually representing the display of exploit information associated with the first subset of suspicious objects differently than the display of exploit information associated with the second subset of the suspicious objects by assigning a specific display location for the exploit information associated with the first subnet of suspicious objects.

7. The threat detection system of claims 3 or 6, wherein the display generation logic visually representing the display of exploit information associated with the first subset of suspicious objects differently than the display of exploit information associated with the second subset of the suspicious objects by modifying a font type of the exploit information associated with the first subset of suspicious objects.

8. The threat detection system of claim 7, wherein the display generation logic visually representing the display of exploit information associated with the first subset of suspicious objects differently than the display of exploit information associated with the second subset of the suspicious objects by modifying a font color of the exploit information associated with the first subset of suspicious objects.

9. The threat detection system of claim 3 or 5, wherein the display generation logic visually representing the display of exploit information associated with the first subset of suspicious objects differently than the display of exploit information associated with the second subset of the suspicious objects by controlling placement of one or more images proximate to the display of exploit information associated with each of the first subset of suspicious objects.

10. The threat detection system of claim 2 further comprising a display generation logic that visually represents a display of exploit information associated with the first subset of suspicious objects differently than a display of exploit information associated with a second subset of the suspicious objects that are received by logic other than the IPS logic by controlling placement of an image with each entry in the display of exploit information that corresponds with one of the first subset of suspicious objects.

11. The threat detection system of claim 7, wherein the display generation logic highlighting the display of the first subset of the suspicious objects by modifying the exploit information associated with the first subset of suspicious objects to appear differently than information associated with exploits associated with the second subset of suspicious objects.

12. The threat detection system of claim 1 comprises a processor and a memory that is communicatively coupled to the processor, the memory comprises the IPS logic and the virtual execution logic.

13. A computerized method comprising:

receiving a first plurality of objects by intrusion protection system (IPS) logic;

filtering the first plurality of objects by the IPS logic that comprises performing either an exploit signature check or a vulnerability signature check on each of the first plurality of objects to identify a second plurality of objects as suspicious objects, the second plurality of objects being a subset of the first plurality of objects and being lesser or equal in number to the first plurality of objects;

automatically verify, by a virtual execution logic, that a first subset of suspicious objects from the second plurality of objects are exploits, the virtual execution logic including at least one virtual machine configured to virtually process content within the suspicious objects and monitor for anomalous behaviors during the virtual processing that are indicative of exploits; and

generating a display that prioritizes information associated with exploits uncovered based on virtual processing of the first subset of suspicious objects.

14. The computerized method of claim 13, wherein the generating of the display comprises prioritizes information associated with verified exploits associated with the first subset of suspicious objects more prominently than information associated with non-verified exploits that are associated with the suspicious objects other than the first subset of suspicious objects.

15. The computerized method of claim 13 or 14, wherein the information associated with the verified exploits comprises: (1) a name of an exploit; (2) a signature pattern used in detection of the exploit; (3) addressing information for a source device that provided the exploit; (4) a time during which an exploit analysis process was conducted; and (5) a name or a version number of software detected to be vulnerable to the exploit.

AMENDED CLAIMS
received by the International Bureau on 01 May 2015 (01.05.2015)

1. A threat detection system, comprising:

an intrusion protection system (IPS) logic configured to receive a first plurality of objects and analyze the first plurality of objects to identify a second plurality of objects as potential exploits, the second plurality of objects being a subset of the first plurality of objects and being lesser or equal in number to the first plurality of objects; and

a virtual execution logic including at least one virtual machine configured to virtually process content within each of the second plurality of objects and monitor for anomalous behaviors during the virtual processing that are indicative of exploits to classify that each object of a first subset of the second plurality of objects is a verified exploit; and

a reporting logic configured to prioritize a visual representation of exploit information associated with the first subset of the second plurality of objects that are classified by the virtual execution logic as verified exploits over exploit information associated with a second subset of the second plurality of objects that are not classified by the virtual execution logic as verified exploits.

2. The threat detection system of claim 1, wherein the reporting logic comprises display generation logic that generates a display of the exploit information associated with the verified exploits differently than a display of exploit information associated with non-verified exploits including the second subset of the second plurality of objects that are not classified by the virtual execution logic as verified exploits.

3. The threat detection system of claim 1 or 2, wherein the IPS logic analyzes the first plurality of objects to identify the second plurality of objects as potential exploits by conducting a signature check operating as either (1) an exploit signature check or (2) a vulnerability signature check,

wherein the exploit signature check compares each of the first plurality of objects to one or more exploit signatures, each of the second plurality of objects is identified as a potential exploit upon matching an exploit signature of the one or more exploit signatures, and wherein the vulnerability signature check compares each of the first plurality of objects to one or more vulnerability signatures, each of the second plurality of objects is identified as a potential exploit upon matching a vulnerability signature that characterizes a sequence of communications that indicate an attempt to attack a software vulnerability protected by the vulnerability signature.

4. The threat detection system of claim 2, wherein the exploit information comprises three or more of the following: (1) a name of an exploit; (2) a signature pattern used in detection of the exploit; (3) addressing information for a source device that provided the exploit; (4) a level of severity of the exploit, where the severity level corresponds, at least in part, to a threat score; (5) a time during which an exploit analysis process was conducted; and (6) a name or a version number of software detected to be vulnerable to the exploit.

5. The threat detection system of claim 2 or 4, wherein the display generation logic visually representing the display of exploit information associated with the verified exploits differently than the display of exploit information associated with the non-verified exploits by assigning a specific display location for the exploit information associated with the verified exploits.

6. The threat detection system of claims 2 or 5, wherein the display generation logic visually representing the display of exploit information associated with the verified exploits differently than the display of exploit information associated with the non-verified exploits by modifying a font type of the exploit information associated with the verified exploits.

7. The threat detection system of claim 2 or 5, wherein the display generation logic visually representing the display of exploit information associated with the verified exploits differently than the display of exploit information associated with the non-verified exploits by modifying a font color of the exploit information associated with the verified exploits.

8. The threat detection system of claim 2 or 7, wherein the display generation logic visually representing the display of exploit information associated with the verified exploits differently than the display of exploit information associated with the

non-verified exploits by controlling placement of one or more images proximate to the display of exploit information associated with each of the verified exploits.

9. The threat detection system of claim 1, wherein the reporting logic comprises a display generation logic that visually represents a display of exploit information associated with the verified exploits differently than a display of exploit information associated with the non-verified exploits that are received from logic other than the IPS logic by controlling placement of an image with each entry in the display of exploit information that corresponds with one of the verified exploits.

10. The threat detection system of claim 2, wherein the reporting logic comprises a classification logic configured to prioritize a visual representation of exploit information associated with the verified exploits over exploit information associated with the non-verified exploits, the classification logic being in communication with the display generation logic that generates the virtual representation of the exploit information associated with the verified exploits and the exploit information associated with the non-verified exploits.

11. The threat detection system of claim 2, wherein the display generation logic highlighting the display of the verified exploits by modifying the exploit information associated with the verified exploits to appear differently than information associated with the second subset of the second plurality of objects that are not classified by the virtual execution logic as verified exploits.

12. The threat detection system of claim 1 comprises a processor and a memory that is communicatively coupled to the processor, the memory comprises at least the IPS logic and the virtual execution logic.

13. A computerized method comprising:

receiving a first plurality of objects by intrusion protection system (IPS) logic;

analyzing the first plurality of objects by the IPS logic that comprises performing either an exploit signature check or a vulnerability signature check on each of the first plurality of objects to identify a second plurality of objects as potential exploits, the second plurality of objects being a subset of the first plurality of objects and being lesser or equal in number to the first plurality of objects;

automatically verify, by a virtual execution logic, that a first subset of the second plurality of objects are exploits, the virtual execution logic including at least one virtual machine configured to virtually process content within the second plurality of objects and monitor for anomalous behaviors during the virtual processing that are indicative of exploits, the second subset of the second plurality of objects that are not classified by the virtual execution logic as verified; and

generating a display that prioritizes information associated with verified exploits over non-verified exploits that comprise a second subset of the second plurality of objects that are not classified by the virtual execution logic as verified exploits.

14. The computerized method of claim 13, wherein the generating of the display comprises prioritizing information associated with verified exploits more prominently than information associated with non-verified exploits.

15. The computerized method of claim 13 or 14, wherein the information associated with the verified exploits comprises: (1) a name of an exploit; (2) a signature pattern used in detection of the exploit; (3) addressing information for a source device that provided the exploit; (4) a time during which an exploit analysis process was conducted; and (5) a name or a version number of software detected to be vulnerable to the exploit.

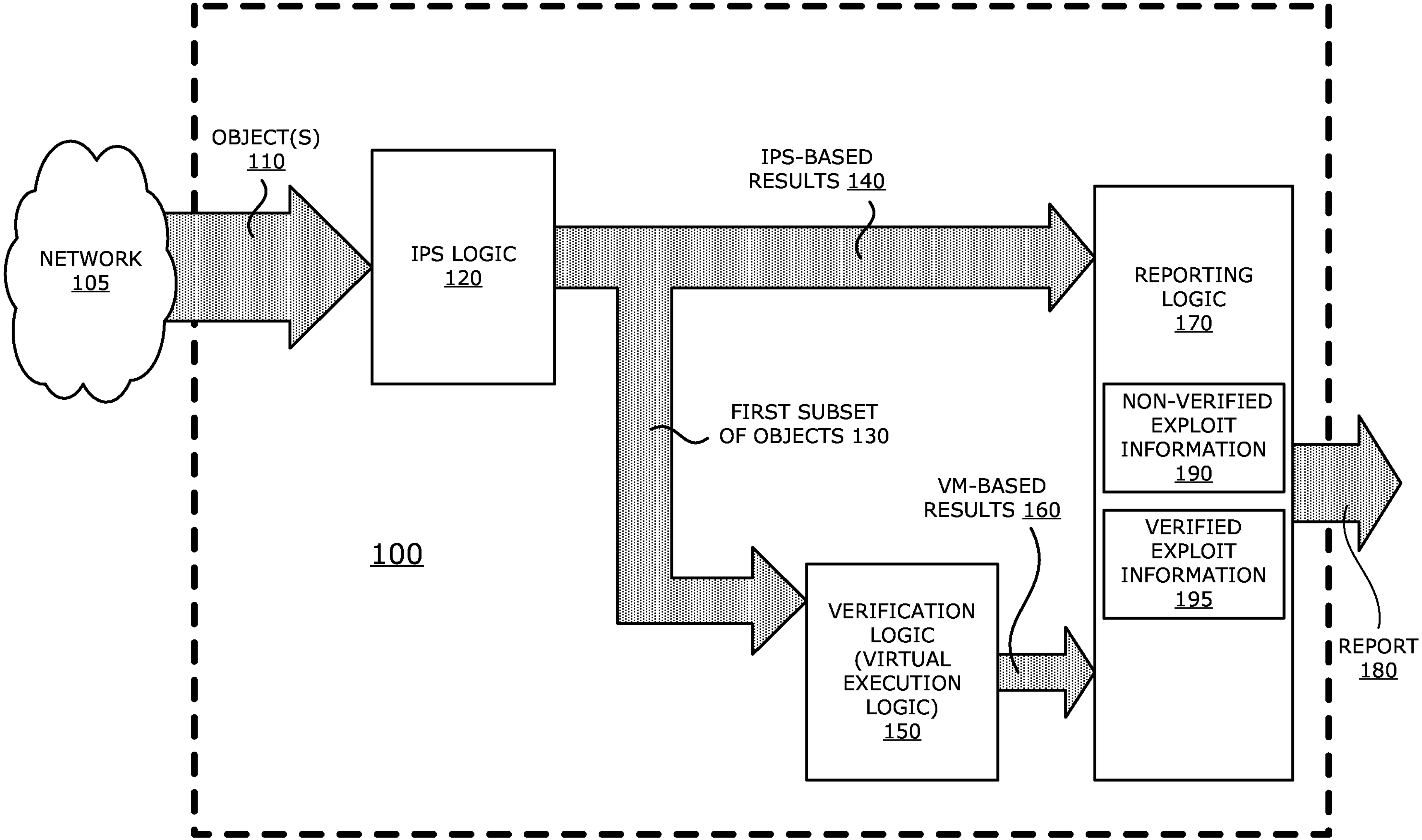

NETWORK
105
OBJECT(S)
110
IPS LOGIC
120
IPS-BASED
RESULTS 140
FIRST SUBSET
OF OBJECTS 130
100
VERIFICATION
LOGIC
(VIRTUAL
EXECUTION
LOGIC)
150
VM-BASED
RESULTS 160
REPORTING
LOGIC
170
NON-VERIFIED
EXPLOIT
INFORMATION
190
VERIFIED
EXPLOIT
INFORMATION
195
REPORT
180

FIG. 1A

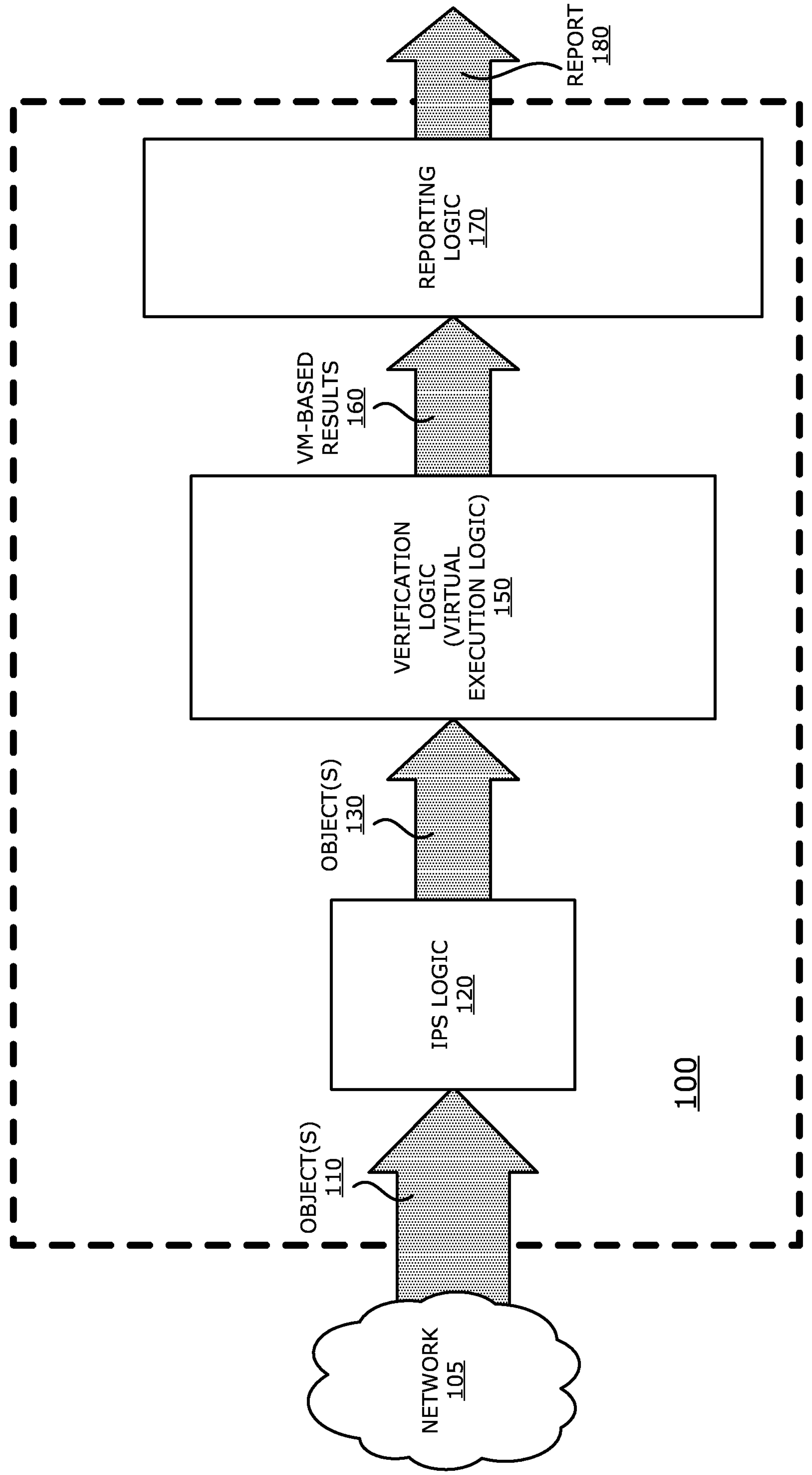

NETWORK
105
OBJECT(S)
110
IPS LOGIC
120
OBJECT(S)
130
VERIFICATION
LOGIC
(VIRTUAL
EXECUTION LOGIC)
150
VM-BASED
RESULTS
160
REPORTING
LOGIC
170
REPORT
180
100

FIG. 1B

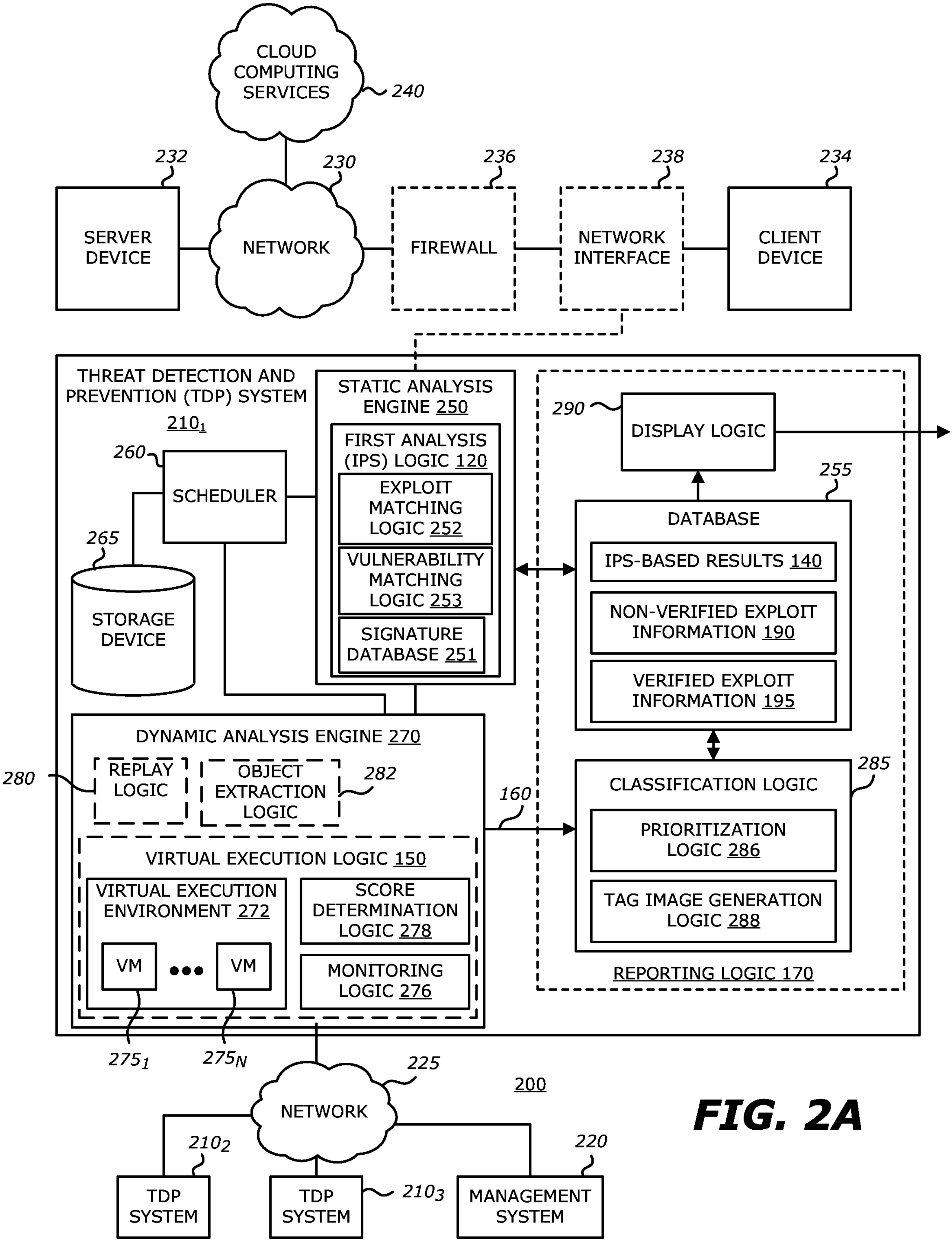

CLOUD COMPUTING SERVICES
240
232
230
236
238
234
SERVER DEVICE
NETWORK
FIREWALL
NETWORK INTERFACE
CLIENT DEVICE
THREAT DETECTION AND PREVENTION (TDP) SYSTEM
210_1
260
SCHEDULER
265
STORAGE DEVICE
STATIC ANALYSIS ENGINE 250
FIRST ANALYSIS (IPS) LOGIC 120
EXPLOIT MATCHING LOGIC 252
VULNERABILITY MATCHING LOGIC 253
SIGNATURE DATABASE 251
290
DISPLAY LOGIC
255
DATABASE
IPS-BASED RESULTS 140
NON-VERIFIED EXPLOIT INFORMATION 190
VERIFIED EXPLOIT INFORMATION 195
285
CLASSIFICATION LOGIC
PRIORITIZATION LOGIC 286
TAG IMAGE GENERATION LOGIC 288
REPORTING LOGIC 170
DYNAMIC ANALYSIS ENGINE 270
280
REPLAY LOGIC
OBJECT EXTRACTION LOGIC
282
160
VIRTUAL EXECUTION LOGIC 150
VIRTUAL EXECUTION ENVIRONMENT 272
VM
VM
SCORE DETERMINATION LOGIC 278
MONITORING LOGIC 276
275_1
275_N
225
NETWORK
200
210_2
210_3
220
TDP SYSTEM
TDP SYSTEM
MANAGEMENT SYSTEM

FIG. 2A

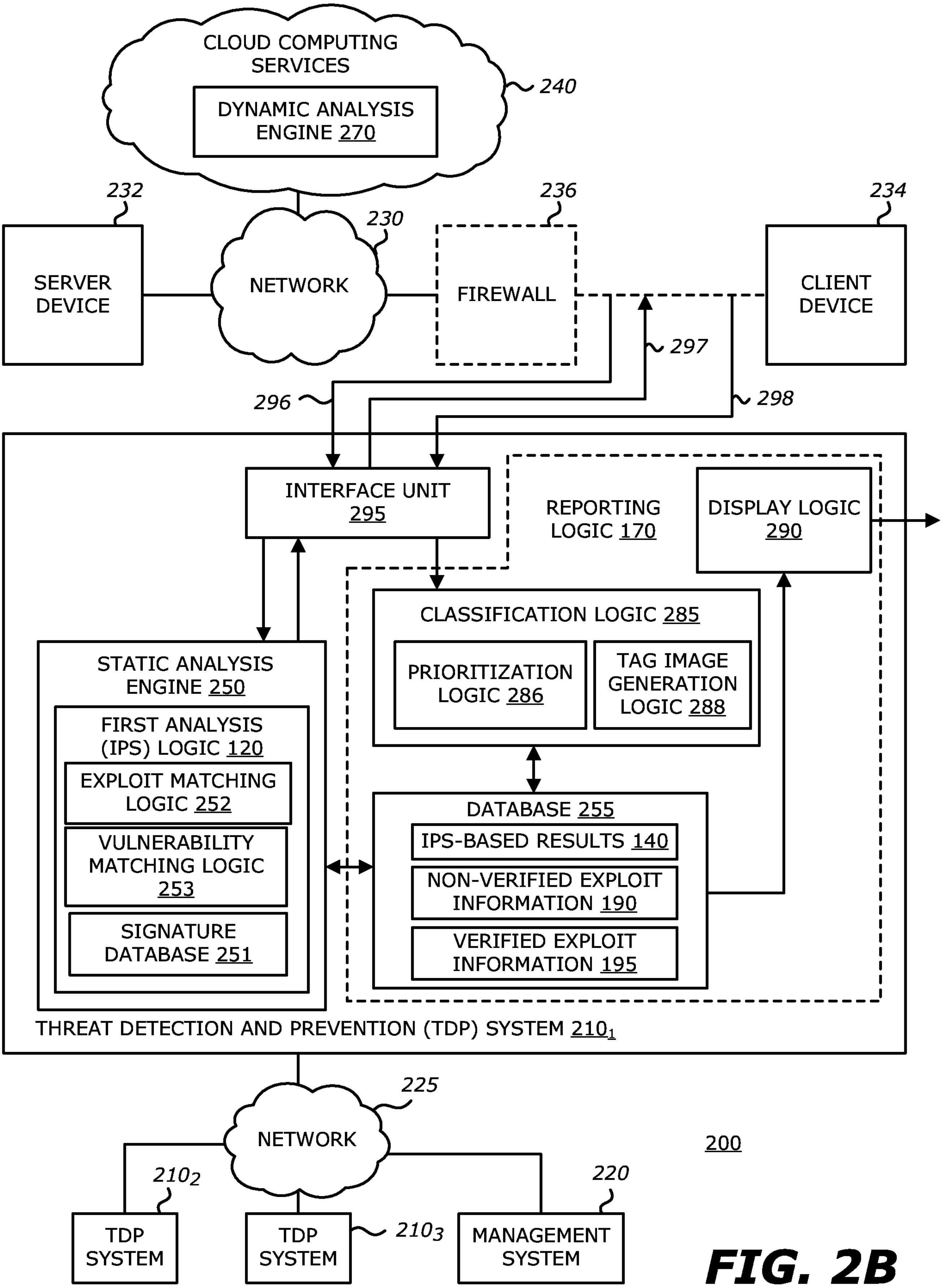

CLOUD COMPUTING SERVICES
DYNAMIC ANALYSIS ENGINE 270
240
232
SERVER DEVICE
230
NETWORK
236
FIREWALL
234
CLIENT DEVICE
297
296
298
INTERFACE UNIT 295
REPORTING LOGIC 170
DISPLAY LOGIC 290
CLASSIFICATION LOGIC 285
PRIORITIZATION LOGIC 286
TAG IMAGE GENERATION LOGIC 288
STATIC ANALYSIS ENGINE 250
FIRST ANALYSIS (IPS) LOGIC 120
EXPLOIT MATCHING LOGIC 252
VULNERABILITY MATCHING LOGIC 253
SIGNATURE DATABASE 251
DATABASE 255
IPS-BASED RESULTS 140
NON-VERIFIED EXPLOIT INFORMATION 190
VERIFIED EXPLOIT INFORMATION 195
THREAT DETECTION AND PREVENTION (TDP) SYSTEM $210_1$
225
NETWORK
200
$210_2$
TDP SYSTEM
TDP SYSTEM
$210_3$
220
MANAGEMENT SYSTEM

FIG. 2B

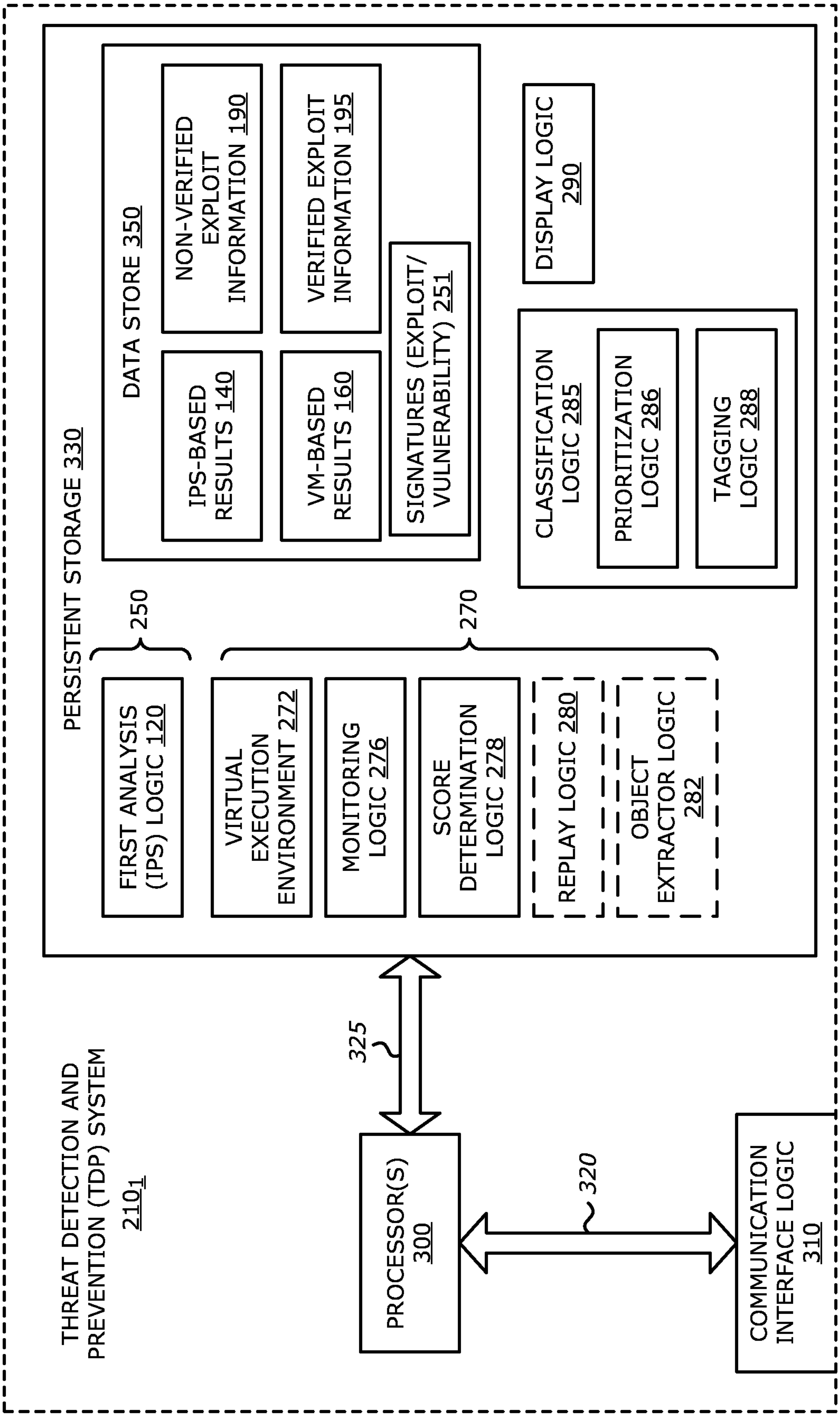

THREAT DETECTION AND PREVENTION (TDP) SYSTEM $210_1$
PERSISTENT STORAGE 330
FIRST ANALYSIS (IPS) LOGIC 120
250
VIRTUAL EXECUTION ENVIRONMENT 272
MONITORING LOGIC 276
SCORE DETERMINATION LOGIC 278
REPLAY LOGIC 280
OBJECT EXTRACTOR LOGIC 282
270
DATA STORE 350
IPS-BASED RESULTS 140
NON-VERIFIED EXPLOIT INFORMATION 190
VM-BASED RESULTS 160
VERIFIED EXPLOIT INFORMATION 195
SIGNATURES (EXPLOIT/ VULNERABILITY) 251
CLASSIFICATION LOGIC 285
PRIORITIZATION LOGIC 286
TAGGING LOGIC 288
DISPLAY LOGIC 290
325
PROCESSOR(S) 300
320
COMMUNICATION INTERFACE LOGIC 310

FIG. 3

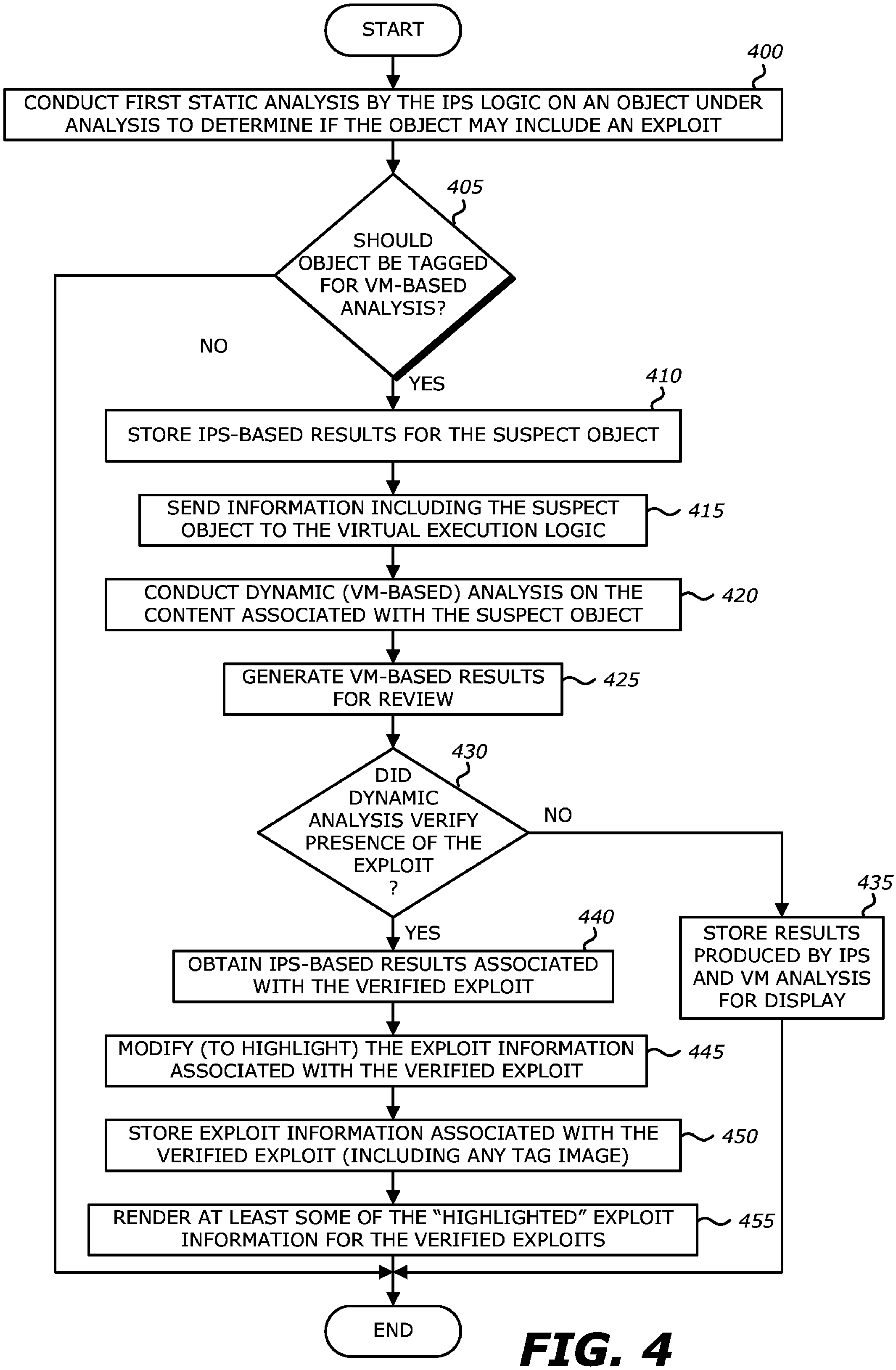
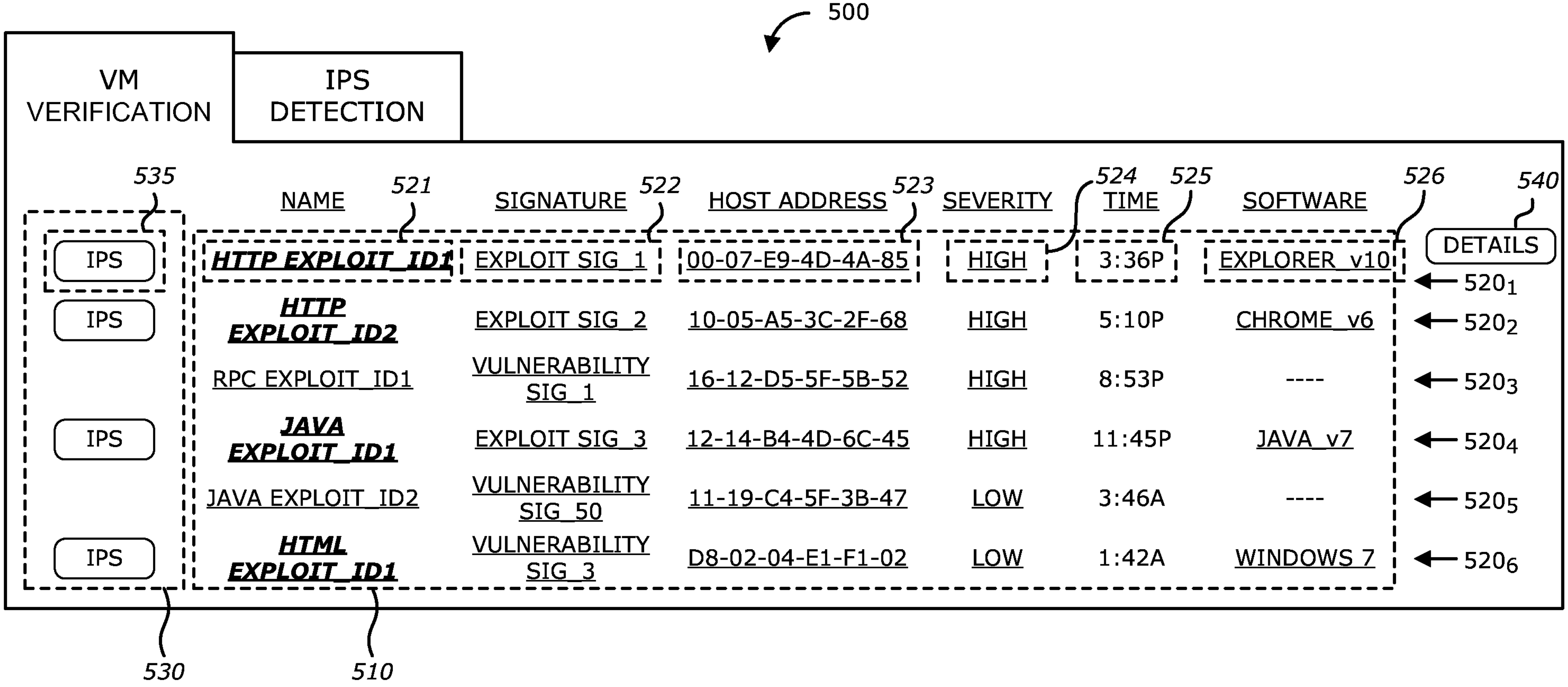
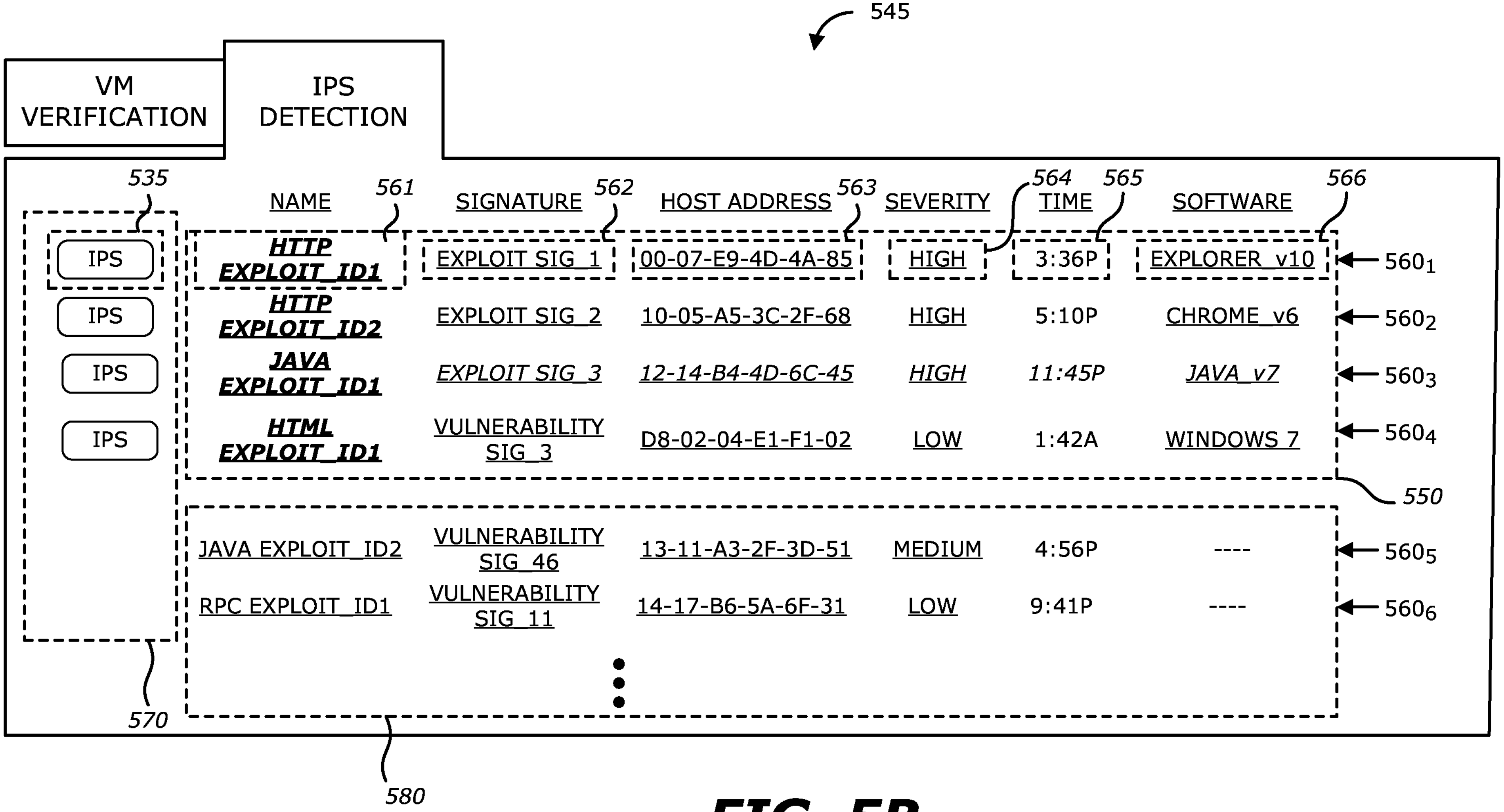

START
400
CONDUCT FIRST STATIC ANALYSIS BY THE IPS LOGIC ON AN OBJECT UNDER ANALYSIS TO DETERMINE IF THE OBJECT MAY INCLUDE AN EXPLOIT
405
SHOULD OBJECT BE TAGGED FOR VM-BASED ANALYSIS?
NO
YES
410
STORE IPS-BASED RESULTS FOR THE SUSPECT OBJECT
SEND INFORMATION INCLUDING THE SUSPECT OBJECT TO THE VIRTUAL EXECUTION LOGIC
415
CONDUCT DYNAMIC (VM-BASED) ANALYSIS ON THE CONTENT ASSOCIATED WITH THE SUSPECT OBJECT
420
GENERATE VM-BASED RESULTS FOR REVIEW
425
430
DID DYNAMIC ANALYSIS VERIFY PRESENCE OF THE EXPLOIT ?
NO
435
STORE RESULTS PRODUCED BY IPS AND VM ANALYSIS FOR DISPLAY
YES
440
OBTAIN IPS-BASED RESULTS ASSOCIATED WITH THE VERIFIED EXPLOIT
MODIFY (TO HIGHLIGHT) THE EXPLOIT INFORMATION ASSOCIATED WITH THE VERIFIED EXPLOIT
445
STORE EXPLOIT INFORMATION ASSOCIATED WITH THE VERIFIED EXPLOIT (INCLUDING ANY TAG IMAGE)
450
RENDER AT LEAST SOME OF THE "HIGHLIGHTED" EXPLOIT INFORMATION FOR THE VERIFIED EXPLOITS
455
END

FIG. 4



| | NAME | SIGNATURE | HOST ADDRESS | SEVERITY | TIME | SOFTWARE |
|---|---|---|---|---|---|---|
| IPS | ***HTTP EXPLOIT_ID1*** | EXPLOIT SIG_1 | 00-07-E9-4D-4A-85 | HIGH | 3:36P | EXPLORER_v10 |
| IPS | ***HTTP EXPLOIT_ID2*** | EXPLOIT SIG_2 | 10-05-A5-3C-2F-68 | HIGH | 5:10P | CHROME_v6 |
| | RPC EXPLOIT_ID1 | VULNERABILITY SIG_1 | 16-12-D5-5F-5B-52 | HIGH | 8:53P | ---- |
| IPS | ***JAVA EXPLOIT_ID1*** | EXPLOIT SIG_3 | 12-14-B4-4D-6C-45 | HIGH | 11:45P | JAVA_v7 |
| | JAVA EXPLOIT_ID2 | VULNERABILITY SIG_50 | 11-19-C4-5F-3B-47 | LOW | 3:46A | ---- |
| IPS | ***HTML EXPLOIT_ID1*** | VULNERABILITY SIG_3 | D8-02-04-E1-F1-02 | LOW | 1:42A | WINDOWS 7 |

FIG. 5A



| NAME | SIGNATURE | HOST ADDRESS | SEVERITY | TIME | SOFTWARE |
|---|---|---|---|---|---|
| ***HTTP EXPLOIT_ID1*** | EXPLOIT SIG_1 | 00-07-E9-4D-4A-85 | HIGH | 3:36P | EXPLORER_v10 |
| ***HTTP EXPLOIT_ID2*** | EXPLOIT SIG_2 | 10-05-A5-3C-2F-68 | HIGH | 5:10P | CHROME_v6 |
| ***JAVA EXPLOIT_ID1*** | *EXPLOIT SIG_3* | *12-14-B4-4D-6C-45* | *HIGH* | *11:45P* | *JAVA_v7* |
| ***HTML EXPLOIT_ID1*** | VULNERABILITY SIG_3 | D8-02-04-E1-F1-02 | LOW | 1:42A | WINDOWS 7 |
| JAVA EXPLOIT_ID2 | VULNERABILITY SIG_46 | 13-11-A3-2F-3D-51 | MEDIUM | 4:56P | ---- |
| RPC EXPLOIT_ID1 | VULNERABILITY SIG_11 | 14-17-B6-5A-6F-31 | LOW | 9:41P | ---- |
|  | ⋮ |  |  |  |  |

FIG. 5B

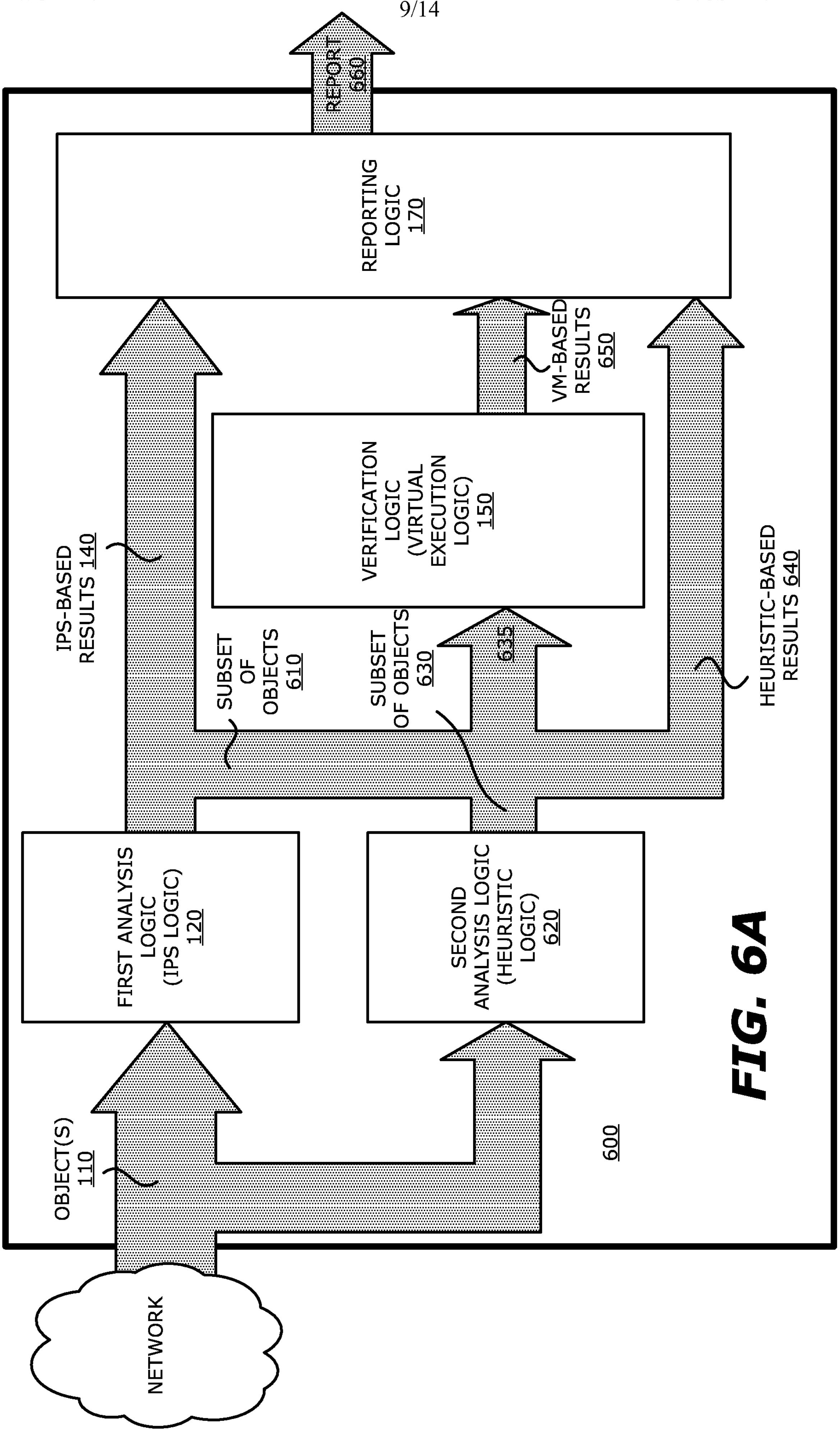

NETWORK
OBJECT(S) 110
FIRST ANALYSIS LOGIC (IPS LOGIC) 120
SECOND ANALYSIS LOGIC (HEURISTIC LOGIC) 620
IPS-BASED RESULTS 140
SUBSET OF OBJECTS 610
SUBSET OF OBJECTS 630
635
HEURISTIC-BASED RESULTS 640
VERIFICATION LOGIC (VIRTUAL EXECUTION LOGIC) 150
VM-BASED RESULTS 650
REPORTING LOGIC 170
REPORT 660
600

FIG. 6A

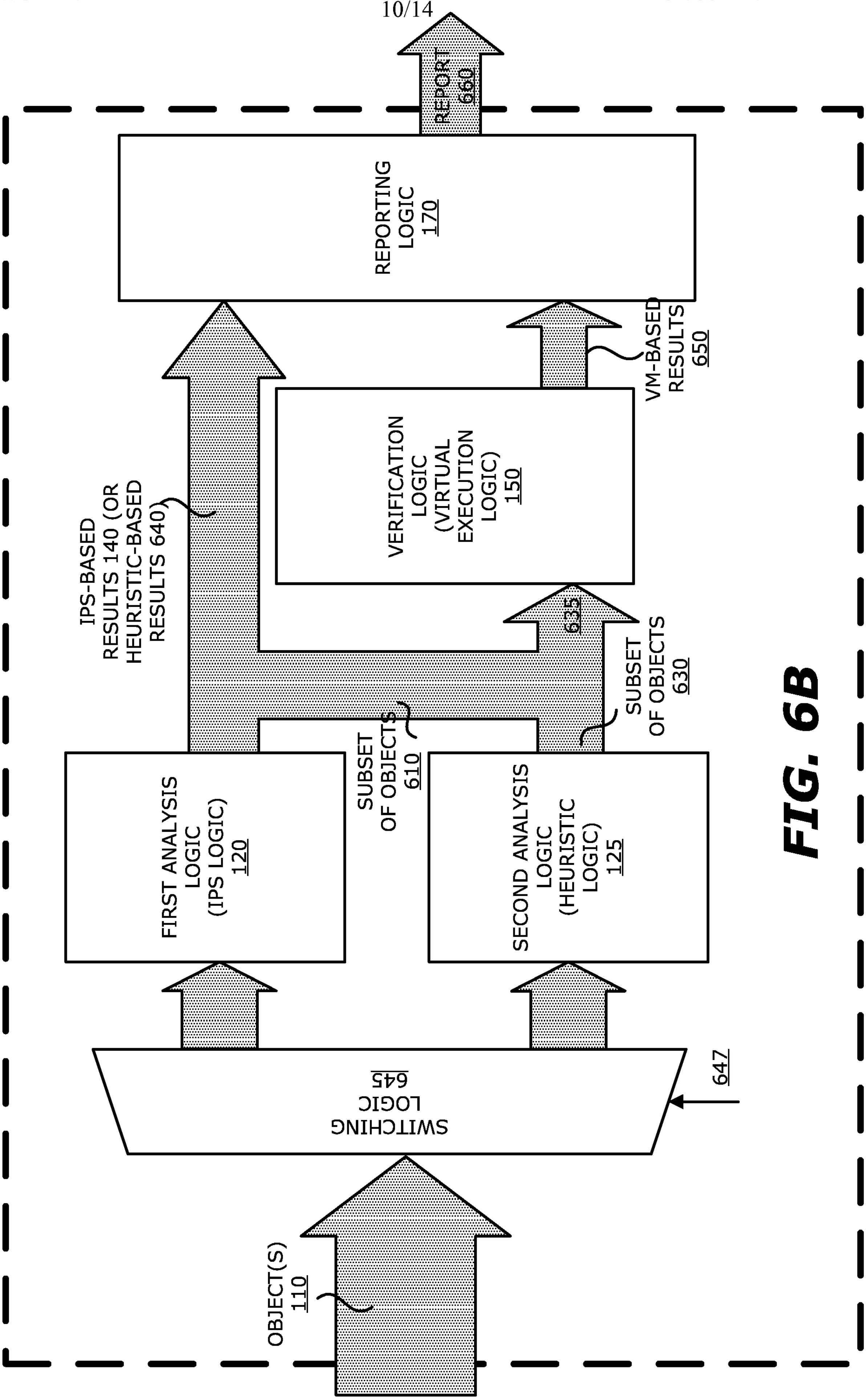

OBJECT(S) 110
SWITCHING LOGIC 645
647
FIRST ANALYSIS LOGIC (IPS LOGIC) 120
SECOND ANALYSIS LOGIC (HEURISTIC LOGIC) 125
SUBSET OF OBJECTS 610
SUBSET OF OBJECTS 630
635
VERIFICATION LOGIC (VIRTUAL EXECUTION LOGIC) 150
IPS-BASED RESULTS 140 (OR HEURISTIC-BASED RESULTS 640)
VM-BASED RESULTS 650
REPORTING LOGIC 170
REPORT 660

FIG. 6B

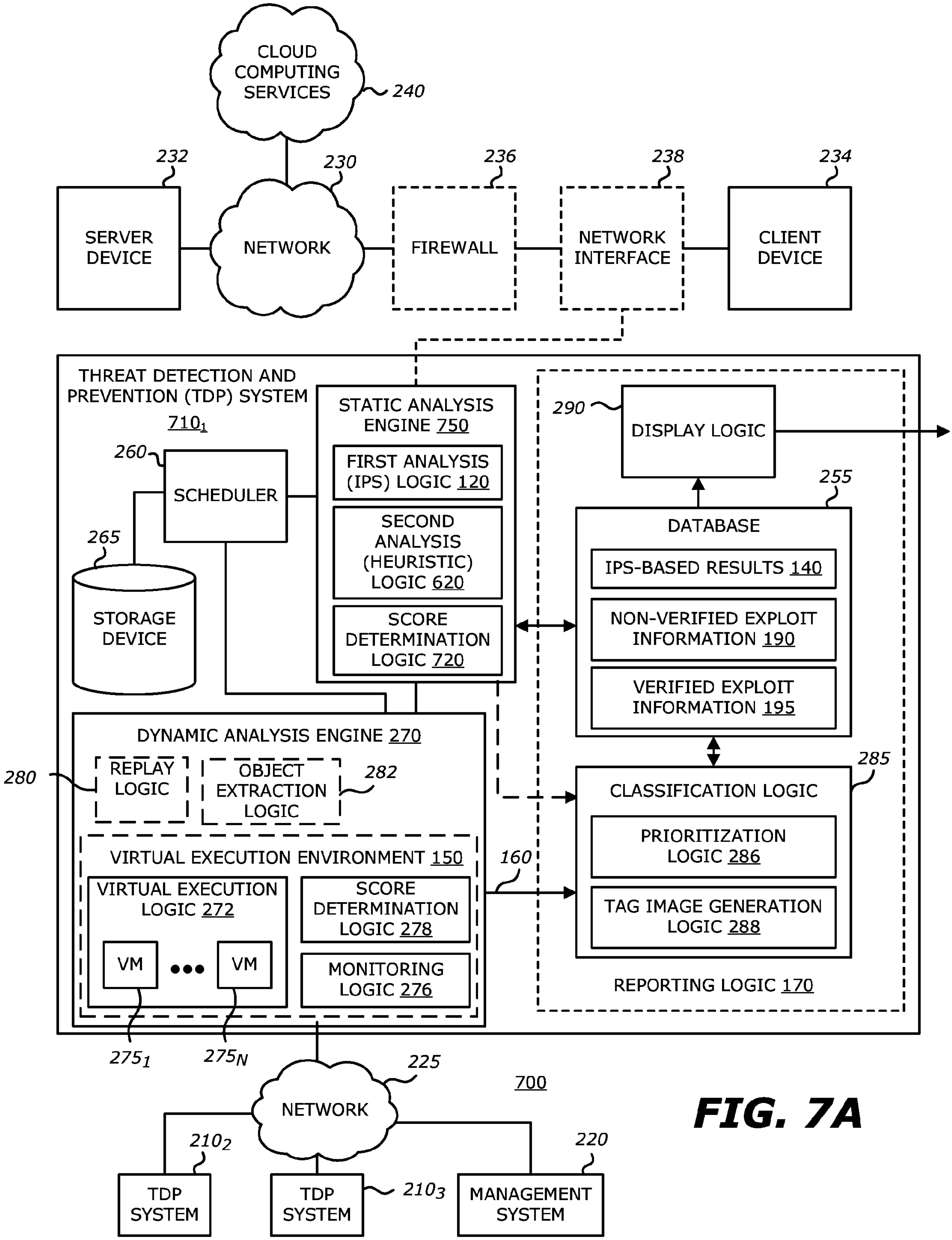

CLOUD COMPUTING SERVICES
240
232
230
236
238
234
SERVER DEVICE
NETWORK
FIREWALL
NETWORK INTERFACE
CLIENT DEVICE
THREAT DETECTION AND PREVENTION (TDP) SYSTEM
$710_1$
STATIC ANALYSIS ENGINE 750
FIRST ANALYSIS (IPS) LOGIC 120
SECOND ANALYSIS (HEURISTIC) LOGIC 620
SCORE DETERMINATION LOGIC 720
290
DISPLAY LOGIC
255
DATABASE
IPS-BASED RESULTS 140
NON-VERIFIED EXPLOIT INFORMATION 190
VERIFIED EXPLOIT INFORMATION 195
260
SCHEDULER
265
STORAGE DEVICE
DYNAMIC ANALYSIS ENGINE 270
280
REPLAY LOGIC
OBJECT EXTRACTION LOGIC
282
285
CLASSIFICATION LOGIC
PRIORITIZATION LOGIC 286
TAG IMAGE GENERATION LOGIC 288
VIRTUAL EXECUTION ENVIRONMENT 150
VIRTUAL EXECUTION LOGIC 272
VM
VM
SCORE DETERMINATION LOGIC 278
MONITORING LOGIC 276
160
REPORTING LOGIC 170
$275_1$
$275_N$
225
NETWORK
700
$210_2$
220
TDP SYSTEM
TDP SYSTEM
$210_3$
MANAGEMENT SYSTEM

FIG. 7A

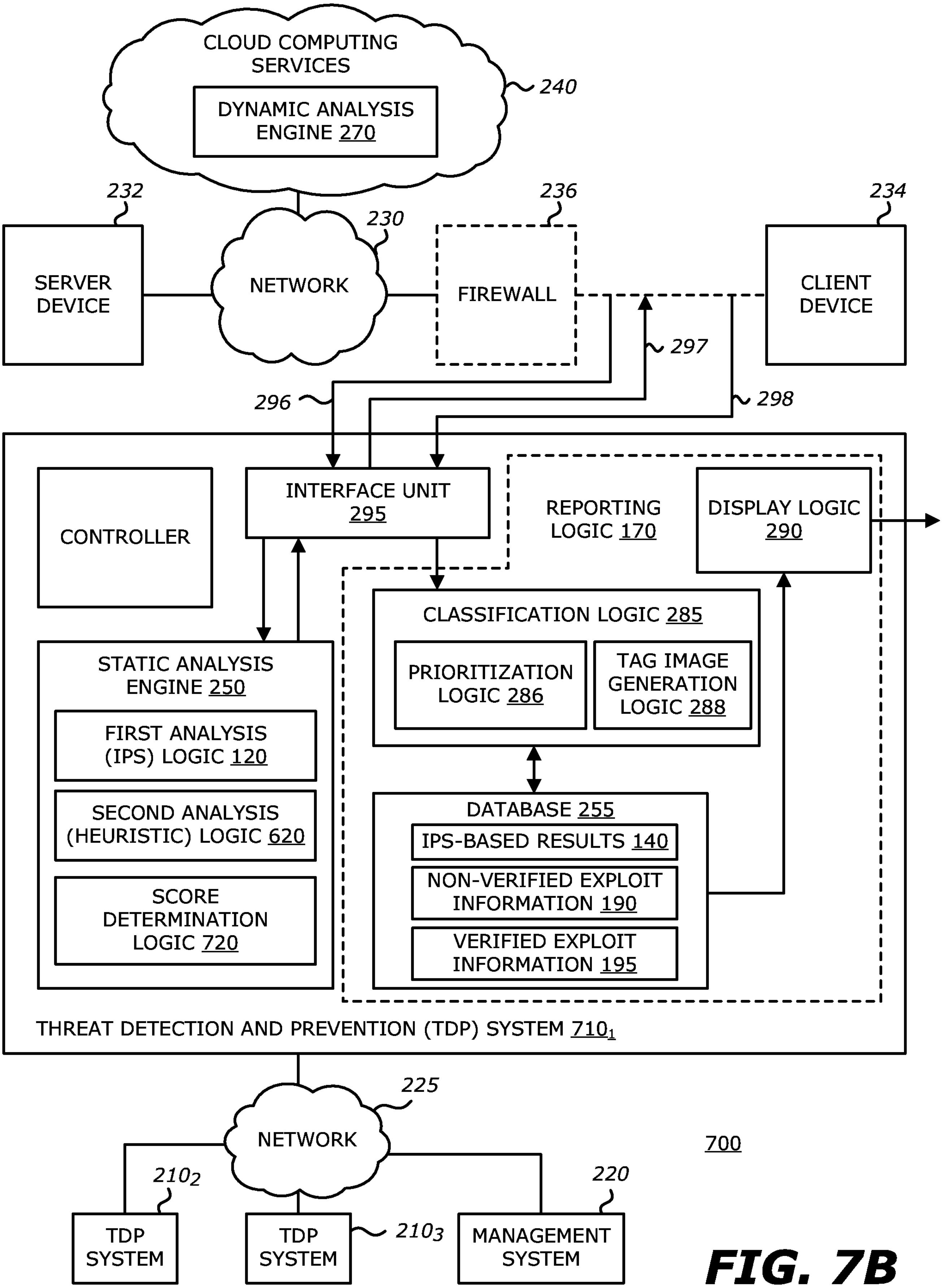

CLOUD COMPUTING SERVICES
DYNAMIC ANALYSIS ENGINE 270
240
232
236
234
230
SERVER DEVICE
NETWORK
FIREWALL
CLIENT DEVICE
297
296
298
CONTROLLER
INTERFACE UNIT 295
REPORTING LOGIC 170
DISPLAY LOGIC 290
CLASSIFICATION LOGIC 285
PRIORITIZATION LOGIC 286
TAG IMAGE GENERATION LOGIC 288
STATIC ANALYSIS ENGINE 250
FIRST ANALYSIS (IPS) LOGIC 120
SECOND ANALYSIS (HEURISTIC) LOGIC 620
SCORE DETERMINATION LOGIC 720
DATABASE 255
IPS-BASED RESULTS 140
NON-VERIFIED EXPLOIT INFORMATION 190
VERIFIED EXPLOIT INFORMATION 195
THREAT DETECTION AND PREVENTION (TDP) SYSTEM $710_1$
225
NETWORK
700
$210_2$
220
TDP SYSTEM
TDP SYSTEM
$210_3$
MANAGEMENT SYSTEM

FIG. 7B

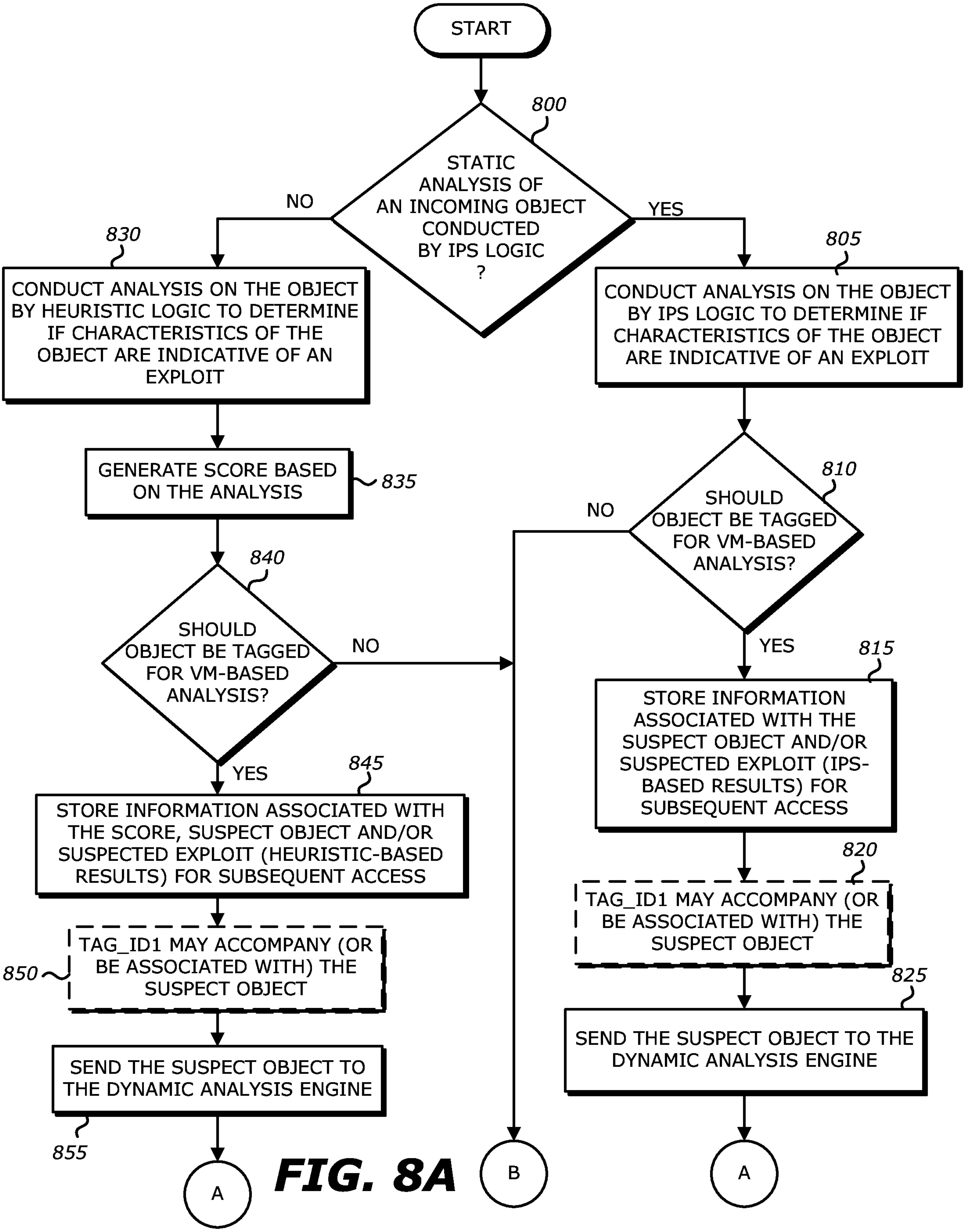

START
800
STATIC ANALYSIS OF AN INCOMING OBJECT CONDUCTED BY IPS LOGIC ?
NO
YES
830
CONDUCT ANALYSIS ON THE OBJECT BY HEURISTIC LOGIC TO DETERMINE IF CHARACTERISTICS OF THE OBJECT ARE INDICATIVE OF AN EXPLOIT
805
CONDUCT ANALYSIS ON THE OBJECT BY IPS LOGIC TO DETERMINE IF CHARACTERISTICS OF THE OBJECT ARE INDICATIVE OF AN EXPLOIT
GENERATE SCORE BASED ON THE ANALYSIS
835
810
NO
SHOULD OBJECT BE TAGGED FOR VM-BASED ANALYSIS?
840
SHOULD OBJECT BE TAGGED FOR VM-BASED ANALYSIS?
NO
YES
815
STORE INFORMATION ASSOCIATED WITH THE SUSPECT OBJECT AND/OR SUSPECTED EXPLOIT (IPS-BASED RESULTS) FOR SUBSEQUENT ACCESS
YES
845
STORE INFORMATION ASSOCIATED WITH THE SCORE, SUSPECT OBJECT AND/OR SUSPECTED EXPLOIT (HEURISTIC-BASED RESULTS) FOR SUBSEQUENT ACCESS
820
TAG_ID1 MAY ACCOMPANY (OR BE ASSOCIATED WITH) THE SUSPECT OBJECT
850
TAG_ID1 MAY ACCOMPANY (OR BE ASSOCIATED WITH) THE SUSPECT OBJECT
825
SEND THE SUSPECT OBJECT TO THE DYNAMIC ANALYSIS ENGINE
SEND THE SUSPECT OBJECT TO THE DYNAMIC ANALYSIS ENGINE
855
A
B
A

FIG. 8A

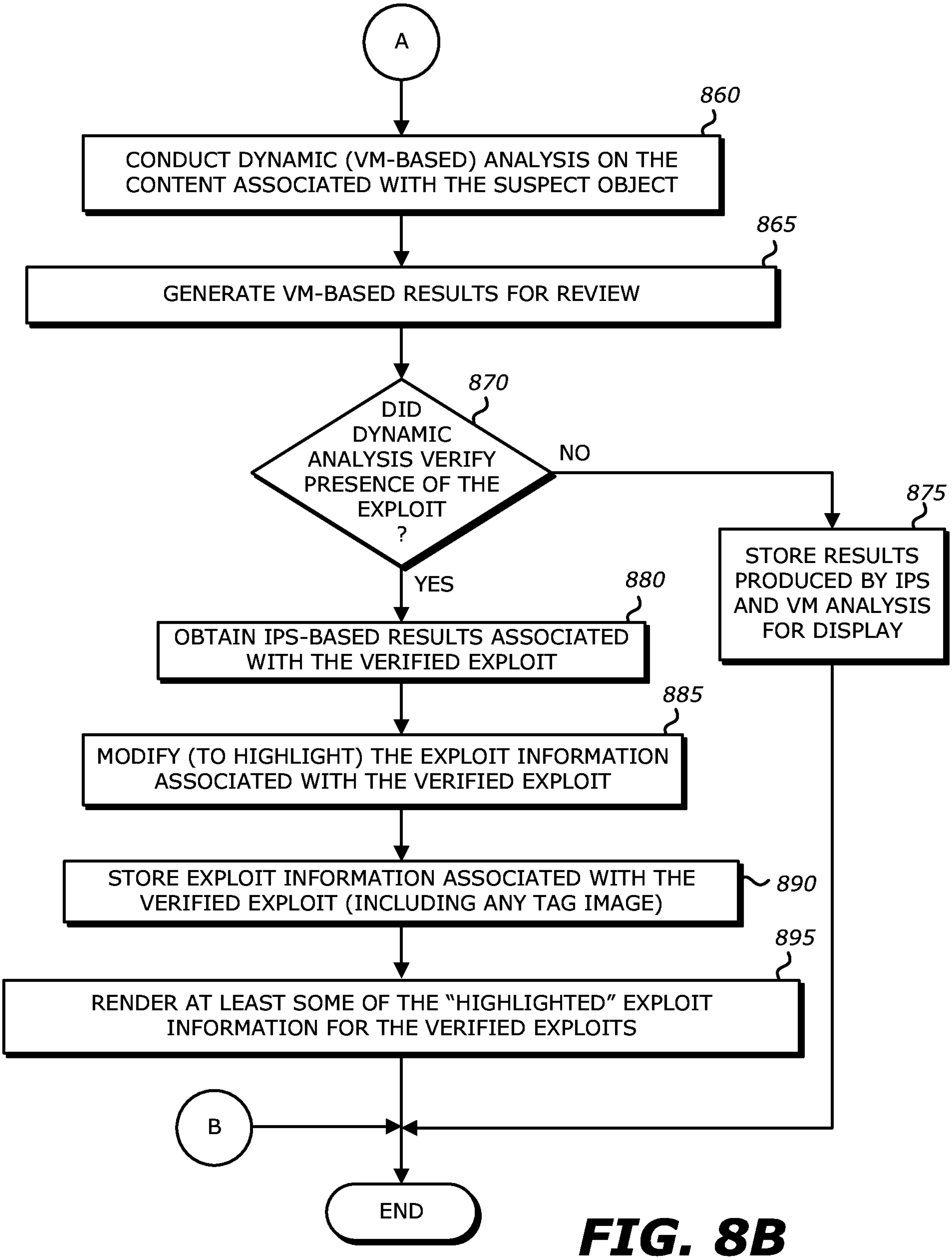

A
860
CONDUCT DYNAMIC (VM-BASED) ANALYSIS ON THE CONTENT ASSOCIATED WITH THE SUSPECT OBJECT
865
GENERATE VM-BASED RESULTS FOR REVIEW
870
DID DYNAMIC ANALYSIS VERIFY PRESENCE OF THE EXPLOIT ?
NO
875
STORE RESULTS PRODUCED BY IPS AND VM ANALYSIS FOR DISPLAY
YES
880
OBTAIN IPS-BASED RESULTS ASSOCIATED WITH THE VERIFIED EXPLOIT
885
MODIFY (TO HIGHLIGHT) THE EXPLOIT INFORMATION ASSOCIATED WITH THE VERIFIED EXPLOIT
890
STORE EXPLOIT INFORMATION ASSOCIATED WITH THE VERIFIED EXPLOIT (INCLUDING ANY TAG IMAGE)
895
RENDER AT LEAST SOME OF THE "HIGHLIGHTED" EXPLOIT INFORMATION FOR THE VERIFIED EXPLOITS
B
END

FIG. 8B

# INTERNATIONAL SEARCH REPORT

International application No
PCT/US2014/072292

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F21/56 H04L29/06
ADD.

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)
H04L G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| X | EP 2 106 085 A1 (HEWLETT PACKARD DEVELOPMENT CO [US])<br>30 September 2009 (2009-09-30)<br>figures 1,2<br>paragraphs [0009], [0016] - paragraph [0017]<br>----- | 1-15 |
| X | US 2007/250930 A1 (AZIZ ASHAR [US] ET AL)<br>25 October 2007 (2007-10-25)<br>figures 7,8,11<br>paragraphs [0013], [0019] - paragraph [0020]<br>paragraph [0046] - paragraph [0047]<br>paragraph [0064] - paragraph [0065]<br>paragraph [0200]<br>claims 1-5<br>-----<br>-/-- | 1-15 |

☒ 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 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<br>23 February 2015 | Date of mailing of the international search report<br>03/03/2015 |
| Name and mailing address of the ISA/<br>European Patent Office, P.B. 5818 Patentlaan 2<br>NL - 2280 HV Rijswijk<br>Tel. (+31-70) 340-2040,<br>Fax: (+31-70) 340-3016 | Authorized officer<br>Martínez Cebollada |

1

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| X | US 8 204 984 B1 (AZIZ ASHAR [US] ET AL)<br>19 June 2012 (2012-06-19)<br>figure 4<br>column 9, line 42 - column 15, line 54<br>----- | 1-15 |
| X | US 2011/321166 A1 (CAPALIK ALEN [US] ET AL) 29 December 2011 (2011-12-29)<br>figures 2-6C<br>paragraphs [0063], [0064]<br>----- | 1-15 |
| X | US 2011/247072 A1 (STANIFORD STUART GRESLEY [US] ET AL)<br>6 October 2011 (2011-10-06)<br>figures 1,3,4<br>page 97 - page 112<br>----- | 1-15 |

1

# INTERNATIONAL SEARCH REPORT

**Information on patent family members**

International application No
PCT/US2014/072292

| Patent document cited in search report | Publication date | Patent family member(s) | Publication date |
|---|---|---|---|
| EP 2106085 A1 | 30-09-2009 | EP 2106085 A1 | 30-09-2009 |
| | | US 2009241190 A1 | 24-09-2009 |
| US 2007250930 A1 | 25-10-2007 | NONE | |
| US 8204984 B1 | 19-06-2012 | NONE | |
| US 2011321166 A1 | 29-12-2011 | AU 2011271157 A1 | 07-02-2013 |
| | | EP 2585965 A1 | 01-05-2013 |
| | | US 2011321166 A1 | 29-12-2011 |
| | | WO 2011163148 A1 | 29-12-2011 |
| US 2011247072 A1 | 06-10-2011 | EP 2666093 A1 | 27-11-2013 |
| | | JP 2014504765 A | 24-02-2014 |
| | | US 2011247072 A1 | 06-10-2011 |
| | | US 2012222121 A1 | 30-08-2012 |
| | | WO 2012100088 A1 | 26-07-2012 |