(19) DANMARK

(10) DK/EP 3588897 T3

(12) Oversættelse af europæisk patentskrift

Patent- og Varemærkestyrelsen

(51) Int.Cl.: ***H 04 L 29/06 (2006.01)***

(45) Oversættelsen bekendtgjort den: **2020-05-25**

(80) Dato for Den Europæiske Patentmyndigheds bekendtgørelse om meddelelse af patentet: **2020-04-22**

(86) Europæisk ansøgning nr.: **18315013.5**

(86) Europæisk indleveringsdag: **2018-06-30**

(87) Den europæiske ansøgnings publiceringsdag: **2020-01-01**

(84) Designerede stater: **AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR**

(73) Patenthaver: **OVH, 2, rue Kellermann, 59100 Roubaix, Frankrig**

(72) Opfinder: **Meriot, Sebastien, 57 Rue Gambetta, 59420 Mouvaux, Frankrig**

(74) Fuldmægtig i Danmark: **Plougmann Vingtoft A/S, Strandvejen 70, 2900 Hellerup, Danmark**

(54) Benævnelse: **FREMGANGSMÅDE OG SYSTEM TIL AT FORSVARE EN INFRASTRUKTUR MOD ET DISTRIBUTED DENIAL OF SERVICE-ANGREB**

(56) Fremdragne publikationer:
JP-A- 2015 222 471
US-B1- 9 356 942
Sébastien Mériot: "Automation Of Internet-of-Things Botnets Takedown By An ISP", , 6 December 2017 (2017-12-06), XP055535591, BotConf 2017 Montpellier Retrieved from the Internet: URL:https://www.botconf.eu/wp-content/uplo ads/2018/01/2017-Meriot-IOT.pdf [retrieved on 2018-12-17]

DESCRIPTION

FIELD

[0001] The present technology relates to the field of Internet security. In particular, the systems and methods for defending an infrastructure against a distributed denial of service attack.

BACKGROUND

[0002] Internet-Of-Things (IoT) is a concept related to the capabilities of physical devices having processing capabilities, for example vehicles, home appliances, wireless routers, printers, security cameras, and other devices, to communicate using the Internet protocol. A growing number of devices have communication capabilities and thus capable of offering services that were unthinkable in the past. Unfortunately, IoT devices are oftentimes not well equipped to resist to threats such as those coming from malwares, a term used to refer to various types of computer viruses and worms. An example of a poorly protected device is a wireless router or a printer for domestic use when the owner has not modified the original security settings of the device.

[0003] Recently, Internet-Of-Things botnets have made the headlines. An IoT device may be infected and become a "bot", a term derived from "robot", when it is infected by a malware. Such malware spreads from peer to peer, i.e. from an IoT device to another, each device becoming a bot. Once a device is infected, it scans the Internet seeking for new vulnerable devices to infect. For some malwares, the infected device sends a copy of the malware to the new device, which in turn becomes infected. For other malwares, once a bot finds a new vulnerable device, it forwards a report to a "reporter" in charge infecting the new device. In any case, the malware spreads quickly to a large number of infected devices. These infected devices are arranged into a "botnet", which is a network of such bots.

[0004] A botnet including a large number of bots can be used to cause a distributed denial of service (DDoS) attack on a computer system, for example a large datacenter. A DDoS attack causes the datacenter to be flooded with superfluous requests. When under such an attack, the datacenter processing and communicating capabilities may become so overloaded that it is temporarily unable to provide service to legitimate users and clients. In at least one event, an attack imposed a load of one (1) terabit per second on an enterprise infrastructure. Because the attack is delivered through a large number of sources, i.e. a large number of bots, having thousands of distinct Internet Protocol (IP) addresses, blocking the attack cannot be achieved by blocking a single source. Given that a DDoS attack may involve such large numbers of IP addresses, some of which being assigned to devices that were legitimate before being infected, a simple blacklisting of potentially harmful IP addresses is not an efficient solution. IP addresses are frequently dynamically assigned, so blacklisting may lead to eventually blocking

legitimate devices.

[0005] Some of the most potent malwares that have been recently discovered include QBOT (also called Bashlite or LizKebab), Mirai (also called Persira, among other variants), Kaiten (also called Tsunami) and Mr. Black. Some originators of malwares have published their source code online, allowing other people to develop modified copies that are not readily discoverable by Internet security software tools. New malwares are placed on line on a continuous basis.

[0006] As expressed hereinabove, the botnet is initially created by infection of legitimate devices, for example IoT devices, by a malware. The bots thus created are made to communicate with a dedicated server, called a Command & Control (C&C) server. Figure 1 (Prior Art) illustrates an example of network in which a C&C server causes a number of legitimate devices to be infected. In a network 10, a C&C server 12 discovers a first set of vulnerable devices, for example a device 14. The device 14 may be accessible without a password or by use of an easily discovered password such as those that are frequently assigned by default by device vendors. The C&C server 12 may also exploit other vulnerabilities of the device 14, for example a "remote code execution" security vulnerability. Having logged into the device 14, the C&C server 12 uploads a malware into the device 14, which becomes infected and turns into a bot. The malware causes the device 14 to search through the Internet 16 for other vulnerable devices 18, 20 and 22. These devices 18, 20 and 22 may for example be accessible via various Transmission Control Protocol (TCP) ports such as, without limitation, ports 22, 23 and 2323. Once the device 14 has identified the vulnerable devices 18, 20 and 22, it sends their respective addresses to the C&C server 12. The C&C server 12 obtains access to the devices 18, 20 and 22 and uploads the malware therein; the devices 18, 20 and 22 turn into bots that also become operable to search for further vulnerable devices. This process explodes exponentially and thousands of vulnerable devices become infected and rapidly become part of the botnet.

[0007] In some instances, the C&C server may be a legitimate server that has been compromised by the actions of a hacker. In some jurisdictions, hosting service providers are legally barred from examining the information from their legitimate clients hosted in their datacenters. This may prevent the operator of the datacenter from easily discovering that one of its own servers is infected and has become a C&C server. The paradox in some cases is that a datacenter may be victim of an attack when the C&C server is part of the datacenter infrastructure.

[0008] Although service providers are oftentimes prevented from scrutinizing the information hosted on their datacenters, this legal requirement preventing a possible manner of mediating the infection of their own servers, they are still responsible for taking appropriate actions to remediate abuse of their service and to thereby continue serving their customers. To this end, abuse remediation solutions may rely on reports sent by third parties, these solutions being too slow to be efficient in many cases.

[0009] Time is often critical when it comes to abuse because customers demand continuous

access to their data and to their services.

[0010] Conventional mitigation solutions are slow to react. Although some solutions have been shown to react within 30 seconds, this delay may actually exceed the duration of a DDoS attack, in which cases the mitigation is essentially fruitless. When the attack imposes a load of one (1) terabit per second on an infrastructure, the damage is very significant, despite the short duration.

[0011] The subject matter discussed in the background section should not be assumed to be prior art merely as a result of its mention in the background section. Similarly, a problem mentioned in the background section or associated with the subject matter of the background section should not be assumed to have been previously recognized in the prior art. The subject matter in the background section merely represents different approaches.

[0012] "Automation Of Internet-of-Things Botnets Takedown by an ISP", by Sébastien Mériot, BotConf 2017 Montpellier, discloses installing a so-called "Honeypot" in an infrastructure in view of attracting a malware. An address or a domain name of the malware is extracted by static or dynamic analysis.

SUMMARY

[0013] Embodiments of the present technology have been developed based on developers' appreciation of shortcomings associated with the prior art.

[0014] In particular, such shortcomings may comprise the delays of current mitigation solutions that may actually exceed the duration of a DDoS attack.

[0015] The object of the invention is solved by a method according to claim 1 and a system according to claim 9. Preferred embodiments are presented in the dependent claims.

[0016] In the context of the present specification, unless expressly provided otherwise, a computer system may refer, but is not limited to, an "electronic device", an "operation system", a "system", a "computer-based system", a "controller unit", a "monitoring device", a "control device" and/or any combination thereof appropriate to the relevant task at hand.

[0017] In the context of the present specification, unless expressly provided otherwise, the expression "computer-readable medium" and "memory" are intended to include media of any nature and kind whatsoever, non-limiting examples of which include RAM, ROM, disks (CD-ROMs, DVDs, floppy disks, hard disk drives, etc.), USB keys, flash memory cards, solid state-drives, and tape drives. Still in the context of the present specification, "a" computer-readable medium and "the" computer-readable medium should not be construed as being the same computer-readable medium. To the contrary, and whenever appropriate, "a" computer-readable medium and "the" computer-readable medium may also be construed as a first

computer-readable medium and a second computer-readable medium.

[0018] In the context of the present specification, unless expressly provided otherwise, the words "first", "second", "third", etc. have been used as adjectives only for the purpose of allowing for distinction between the nouns that they modify from one another, and not for the purpose of describing any particular relationship between those nouns.

[0019] Implementations of the present technology each have at least one of the above-mentioned object and/or aspects, but do not necessarily have all of them. It should be understood that some aspects of the present technology that have resulted from attempting to attain the above-mentioned object may not satisfy this object and/or may satisfy other objects not specifically recited herein.

[0020] Additional and/or alternative features, aspects and advantages of implementations of the present technology will become apparent from the following description, the accompanying drawings and the appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0021] For a better understanding of the present technology, as well as other aspects and further features thereof, reference is made to the following description which is to be used in conjunction with the accompanying drawings, where:

Figure 1 (Prior Art) illustrates an example of network in which a C&C server causes a number of legitimate devices to be infected;

Figure 2 is a block diagram of an infrastructure implementing a method and a system for defending against a DDoS attack in accordance with an embodiment of the present technology;

Figure 3 is a high-level flowchart representing operations of a method for defending an infrastructure against a DDoS attack in accordance with an embodiment of the present technology;

Figure 4 is a high-level flowchart representing operations of an automated reverse engineering method for extracting an address or a domain name of a C&C server in accordance with an embodiment of the present technology;

Figures 5A and 5B are a sequence diagram showing detailed operations of the method for defending an infrastructure against a DDoS attack in accordance with an embodiment of the present technology;

Figure 6 is an actual example of a log of commands detected at the infrastructure;

Figure 7 is an illustration of a routine used by the malware MIRAI to encrypt character chains;

Figure 8 is a sequence diagram showing detailed operations of the reverse engineering method for extracting an address or a domain name of a C&C server in accordance with an embodiment of the present technology;

Figure 9 is a block diagram of a computer platform in accordance with an embodiment of the present technology;

Figure 10 is a graph showing a variation of a number of abuse notifications in the infrastructure before and after implementation the present technology;

Figure 11 is a graph showing a variation of a number of C&C servers hosted in the infrastructure before and after implementation the present technology; and

Figure 12 is a graph showing a worldwide variation of a number of infections in the same timescale as in Figure 11.

[0022] It should also be noted that, unless otherwise explicitly specified herein, the drawings are not to scale.

DETAILED DESCRIPTION

[0023] The examples and conditional language recited herein are principally intended to aid the reader in understanding the principles of the present technology and not to limit its scope to such specifically recited examples and conditions.

[0024] Furthermore, as an aid to understanding, the following description may describe relatively simplified implementations of the present technology. As persons skilled in the art would understand, various implementations of the present technology may be of a greater complexity.

[0025] In some cases, what are believed to be helpful examples of modifications to the present technology may also be set forth. This is done merely as an aid to understanding, and, again, not to define the scope or set forth the bounds of the present technology. These modifications are not an exhaustive list, and a person skilled in the art may make other modifications while nonetheless remaining within the scope of the present technology. Further, where no examples of modifications have been set forth, it should not be interpreted that no modifications are possible and/or that what is described is the sole manner of implementing that element of the present technology.

[0026] Moreover, all statements herein reciting principles, aspects, and implementations of the present technology, as well as specific examples thereof, are intended to encompass both structural and functional equivalents thereof, whether they are currently known or developed in

the future. Thus, for example, it will be appreciated by those skilled in the art that any block diagrams herein represent conceptual views of illustrative circuitry embodying the principles of the present technology. Similarly, it will be appreciated that any flowcharts, flow diagrams, state transition diagrams, pseudo-code, and the like represent various processes which may be substantially represented in computer-readable media and so executed by a computer or processor, whether or not such computer or processor is explicitly shown.

[0027] The functions of the various elements shown in the figures, including any functional block labeled as a "processor", may be provided through the use of dedicated hardware as well as hardware capable of executing software in association with appropriate software. When provided by a processor, the functions may be provided by a single dedicated processor, by a single shared processor, or by a plurality of individual processors, some of which may be shared. In some embodiments of the present technology, the processor may be a general purpose processor, such as a central processing unit (CPU) or a processor dedicated to a specific purpose, such as a digital signal processor (DSP). Moreover, explicit use of the term a "processor" should not be construed to refer exclusively to hardware capable of executing software, and may implicitly include, without limitation, application specific integrated circuit (ASIC), field programmable gate array (FPGA), read-only memory (ROM) for storing software, random access memory (RAM), and non-volatile storage. Other hardware, conventional and/or custom, may also be included.

[0028] Software modules, or simply modules which are implied to be software, may be represented herein as any combination of flowchart elements or other elements indicating performance of process steps and/or textual description. Such modules may be executed by hardware that is expressly or implicitly shown. Moreover, it should be understood that module may include for example, but without being limitative, computer program logic, computer program instructions, software, stack, firmware, hardware circuitry or a combination thereof which provides the required capabilities.

[0029] With these fundamentals in place, we will now consider some non-limiting examples of systems and methods adapted to defend an infrastructure against a distributed denial of service (DDoS) attack.

Infrastructure

[0030] Referring now to the drawings, Figure 2 is a block diagram of an infrastructure implementing a method and a system for defending against a DDoS attack in accordance with an embodiment of the present technology. An infrastructure 100 may for example represent a data center, or a plurality of data centers, providing hosting services for one or more customers. In the example of Figure 2, a game server 110 owned by a customer of the infrastructure operator is hosted in the infrastructure 100. It will be appreciated that the infrastructure 100 may include a large number of servers for hosting services for a large number of customers and that the infrastructure 100 may be distributed over a plurality of

datacenters (not shown) for redundancy, reliability and/or load sharing purposes. The datacenters forming the infrastructure 100 may be geographically distributed, for example worldwide. The illustrated infrastructure 100 of Figure 2 is heavily simplified for ease of illustration.

[0031] For defensive purposes, specific applications are installed on computer platforms (shown on a later Figure) of the infrastructure 100, for example computers or servers, to form a system for defending the infrastructure 100 against DDoS attacks. One such application is a software decoy 120, sometimes called a "honeypot", which is configured to pose as a vulnerable device that could easily be infected by a malware. The software decoy 120 has simple login credentials that mimic those that are generally conferred by default to simple devices. For example, the login credentials may include a login identity such as "root" and a password that is simply set to "password". The software decoy 120 may also attempt to detect remote code execution (RCE) attacks. The software decoy 120 is intended to download a malware but to refrain from installing the malware. Another application of the system is a C&C data collector 130 configured to extract information from the malware received at the software decoy 120. A further application of the system is a client 140 that is configured to pose as an infected device. [[,]] Yet another application of the system is a cleaning component 150. Although a single software decoy 120, a single C&C data collector 130, a single client 140 and a single cleaning component 150 are shown, any one of these components may be replicated into a plurality of components in the infrastructure 100.

[0032] In an embodiment, all components of the infrastructure 100 may be co-located in a same installation, for example being part of a same datacenter. In another embodiment, some of the components of the infrastructure 100 may be distant from other components of the infrastructure 100, geographically separated components of the infrastructure communicating via the Internet or via a private network. In an illustrative but non-limiting example, servers providing services to customers of the infrastructure, for example the game server 110, may be installed in large numbers in many locations while, in the same example, components of the system for defending against DDoS attacks may be installed in a limited number of locations.

[0033] A server, which is usually but not necessarily outside of the infrastructure 100, is the command and control (C&C) server 12 of Figure 1. The C&C server 12 may either be a server of a criminal entity or a legitimate server that has been infected by a malware. The C&C server 12 causes a device, which is usually legitimate but poorly protected, to become infected by the malware, turning the device into bots such as the bots 18, 20 and 22 of Figure 1. Although just a few bots are illustrated, the infrastructure 100 may be subject to an attack from a botnet including thousands of bots. It is usually one of the bots 18, 20 and 22 that discover the software decoy 120 and attempt to infect it with the malware.

High level processes

[0034] Figure 3 is a high-level flowchart representing operations of a method for defending an

infrastructure against a DDoS attack in accordance with an embodiment of the present technology. A flowchart 200 starts with a scan 210 that actually take place in the C&C server 12 or a network (i.e. a botnet) formed by the bots 18, 20 and 22. The scan 210 is used in an attempt to find vulnerable devices in the Internet. The software decoy 120 (the "honeypot") is discovered during the scan 210. One of the bots 18. 20 or 22 successfully logs on the software decoy 120. Then, a challenge 220 is send to the software decoy 120; this challenge is sent in view of the fact that malware authors are aware of the concept of using honeypots and attempt to verify whether the device having responded to the scan 210 is in fact a honeypot. The software decoy 120 is configured to provide a response 230 to the challenge that corresponds to the expected response that a vulnerable device would actually send. For instance, the challenge may include a request for displaying the content of a file, or to display a hexadecimal character chain. The software decoy 120 replies accordingly. In a variant, the infrastructure 100 may include a plurality of software decoys 120 (i.e. a plurality of honeypots) that are each equipped to provide different challenge responses so that at least one of the software decoy 120 not identified as a honeypot by the attacker. The remainder of the flowchart 200 is based on the assumption that the botnet does not detect the true nature of the software decoy 120. A command carrying an Internet Protocol (IP) address or a uniform resource locator (URL) of a downloader of the malware for fetching and downloading the malware therefrom is received by the software decoy 120 at operation 240; this is illustrated on Figure 3 as a honeypot recovering a sample software that is, in fact, the malware. This address is frequently the address of a web server intended to deliver the malware to vulnerable devices, although it may also be the address of the C&C server 12. The software decoy 120 downloads the malware and receives a command requesting installing the malware. The software decoy 120 does not actually install the malware. Following operations of the flowchart 200 include an analysis operation 250 of the malware by the C&C data collector 130 in order to recover, at operation 260, the address and the port number of the C&C server 12 or the domain name and the port number of the C&C server 12. A connection 270 is made to the C&C server 12 by the client 140, using this address. This connection operation 270 allows the client 140 to obtain further information about the C&C server 12, including without limitation particulars of an upcoming DDoS attack. The client 140 informs other components, for example the cleaning component 150, of the particular of the upcoming DDoS attack. If the address of the C&C server 12 is found to be part of the infrastructure 100, an "abuse" notification is forwarded at operation 280. The term "abuse" is used in this context because it is discovered that a server that is part of the infrastructure 100 has been infected to become the C&C server 12. Abuse mitigation procedures are initiated at operation 290. The abuse mitigation procedures may include a quarantine or a clean-up of the infected device of the infrastructure 100 turned into the C&C 12.

[0035] Figure 4 is a high-level flowchart representing operations of an automated reverse engineering method for extracting an address or a domain name of a C&C server in accordance with an embodiment of the present technology. A flowchart 300 essentially corresponds to operations 250 and 260 of Figure 3. At operation 310, the malware may be unpacked, if the malware is packed when received, and a first ciphering key of a group of known ciphering keys (also called encryption keys) of a previously detected malware is applied

thereto, using an exclusive-OR (XOR) operation, in an attempt to decipher the malware. A static analysis of the malware is made at operation 320 in an attempt to recover the address or domain name of the C&C server 12; this attempt may fail if the first ciphering key applied at operation 310 is not appropriate. If so, operation 310 is executed again using another known ciphering key. If none of the know ciphering keys applied at repeated operations 310 leads to successfully deciphering the malware at repeated operations 320, a dynamic analysis is attempted at operation 330. Details of this analysis will be described hereinbelow in relation to Figure 8.

Detailed defensive process

[0036] Figures 5A and 5B are a sequence diagram showing detailed operations of the method for defending an infrastructure against a DDoS attack in accordance with an embodiment of the present technology. On Figures 5A and 5B, a sequence 400 for defending an infrastructure against a DDoS attack comprises a plurality of operations that may be executed in variable order, some of the operations possibly being executed concurrently, some of the operations being optional. The sequence 400 shows operations that take place in various components of the infrastructure 100.

[0037] The sequence 400 is initiated at operation 405 by installing the software decoy in the infrastructure. As expressed hereinabove, the software decoy 120 is configured to pose as a vulnerable device. The software decoy 120 may receive, at operation 410, a challenge intended to detect a protection function of the infrastructure; otherwise stated, the challenge is intended to determine whether the software decoy 120 that has been located by the botnet functions as a honeypot. The challenge may be received from a bot. If the software decoy 120 detects that a source of the challenge is located in the infrastructure 100, for example when the challenge has a source IP address hosted in the infrastructure 100, the software decoy 120 may issue an abuse notification (operation 280 of Figure 3). The software decoy forwards a challenge response at operation 415. Usually, this challenge response will suffice to hide the true nature of the software decoy 120 to the botnet. As expressed earlier, the infrastructure may include a plurality of software decoys 120 in the hope that at least one challenge response will trick the botnet in hiding the honeypot function of the software decoy 120. Implementing a large number of software decoy instances in the infrastructure 100 increases the probability that the botnet will rapidly discover one of the software decoys.

[0038] At operation 420, a malware intended to infect the software decoy 120 is received at the software decoy 120. A downloader of the malware may for example be a compromised web server. In at least one embodiment, the software decoy 120 does not actually install the malware. If the software decoy 120 detects that the downloader of the malware is located in the infrastructure 100, for example when the malware has a source IP address or a uniform resource locator (URL) hosted in the infrastructure 100, the software decoy 120 may issue an abuse notification. The software decoy 120 forwards the malware to the C&C data collector 130 that extracts therefrom an address or a domain name of the C&C server 12 at operation

425. If the C&C data collector 130 detects that the address of the domain name of the C&C server 12 belongs to the infrastructure 100, the C&C data collector 130 may issue an abuse notification. The address of the C&C server may be an IP address. The domain name of the C&C server may be part of a URL. A non-limiting example embodiment of operation 425 as performed by the C&C data collector 130 is provided hereinbelow, in the description of Figure 8.

[0039] Using the address or the domain name of the C&C server 12, the client 140 connects to the C&C server 12 at operation 430. As expressed hereinabove, the client 140 is configured to pose as being infected by the malware. To this end, the client 140 may implement a known protocol of the malware. At operation 435, the client 140 receives a command intended by the C&C server 12 to cause the client 140 to participate in the DDoS attack. The connection between the client 140 and the C&C server 12 may be lost and automatically reinstated, in which case the client 140 may verify again the address or the domain name of the C&C server 12 in view of issuing an abuse notification, if applicable.

[0040] The particulars of the DDoS attack may comprise an address of an intended victim of the DDoS attack, for example the game server 110, a port number of the intended victim of the DDOS attack, an intended duration of the DDoS attack, and/or similar parameters. For example, Figure 6 is an actual example of a log of commands detected at the infrastructure 100. On Figure 6, actual IP addresses of legitimate systems, except for the least significant digits, are masked by hiding actual values with "x values" to protect the anonymity of actual owners of these addresses. A command detected on June 26, at 16:47:40, showed that a standard (STD) DDoS attack, actually a TCP flood, was commanded by the C&C server 12 to bots of the botnet. The DDoS attack was directed to IP address "xx.xxx.xxx.42", at port number 443, and had an intended duration of 200 seconds.

[0041] Returning to Figure 5B, the client 140 forwards particulars of the DDoS attack to the cleaning component 150 at operation 440. In a variant, forwarding the particulars of the DDoS attack to the cleaning component 150 may be conditional to the intended victim being part of the infrastructure 100, as in the case of the game server 110 and as determined based on an address being par to the particulars of the DDOS attack. However, another variant in which the particulars of the DDoS attack are forwarded to another system or apparatus, for example another infrastructure, may also be contemplated; this could be the case, for example, when the operator and/or owner of the infrastructure 100 has an agreement with the operator of the other system or apparatus for cooperating in the prevention of DDoS attacks.

[0042] The infrastructure 100 may comprise a routing table used to direct incoming signals, messages and packets to an appropriate component of the infrastructure 100. In an embodiment, operation 445 comprises updating the routing table of the infrastructure 100 to cause routing of incoming messages destined to the address of the intended victim toward the cleaning component 150 when the intended victim is part of the infrastructure, as in the case of the game server 110. The border gateway protocol (BGP) or the open shortest path first (OSPF) protocol may be used to update the routing table. When the infrastructure 100 detects

that the DDoS attack has ended, the routing table may recover its previous state. Any component of the infrastructure 100 may initiate operation 445.

[0043] At operation 450, the cleaning component 150 discards incoming signals having at least one of the particulars of the DDoS attack. In a non-limiting example, the particulars of the DDoS attack include an IP address and/or a port number of the intended victim and the cleaning component 150 may filter incoming signals, messages and/or packets having this IP address in the IP header and/or having this port number in their Transmission Control Protocol (TCP) header.

[0044] Operation 450 may include one or more of sub-operations 451, 452 and/or 453. At sub-operation 451, incoming signals carrying spoofed source IP addresses are discarded. At sub-operation 452, incoming signals that are not related to previously established connections are discarded. At sub-operation 453, incoming signals are discarded when a number of such signals from the same source IP address exceeds a predetermined threshold. Sub-operations 451, 452 and 453 represent non-limiting example embodiments of the operation 450 and the cleaning component 150 may apply other criteria for discarding incoming signals having at least one of the particulars of the DDoS attack.

[0045] If a bot, the downloader of the malware or the C&C server 12 is found to be hosted in the infrastructure 100, an abuse of the resources of the infrastructure 100 has been discovered. A customer of the infrastructure 100 may have content hosted in a server or other component of the infrastructure 100 that is compromised by the installation of the bot, of the downloader of the malware, or of the C&C server 12 in that compromised component. At operation 455, the customer may be alerted. Alternatively or in addition, the compromised component of the infrastructure 100 may be placed in quarantine. Any component of the infrastructure 100 may initiate operation 455.

Reverse engineering

[0046] The present technology introduces several variants for extracting the address or domain name of the C&C server 12 that controls the botnet. Sometimes, the address or domain name may be contained in a character string of the malware that is not encrypted, for example in the case of the QBOT malware. In such cases, the extraction of the domain name or of the C&C server 12 is trivially done by the C&C data collector 130.

[0047] More frequently, the address or domain name and port number are hidden by encryption within the malware. Reverse engineering may be in these cases be used to uncover the address or domain name of the C&C server 12. Most malwares can be recognized by locating in a signature in their binary, the signature comprising for example a particular code sequence. For example, the malware MIRAI hides an IP address of the C&C 12 in an unsigned 32-bit integer value associated with machine language instructions dedicated to manipulating addresses. In the particular, non-limiting example of the Intel™ x86 instruction set, examination

of the MOV and PUSH operation codes may reveal the IP address and a port number of the C&C server 12. When using other architectures, for example those using ARM™, MIPS™ or SPARC™ processors, other operation codes may be examined for the same purposes.

[0048] In a variant, reverse engineering uses a static analysis of the binary of the malware and automatically searches the address or domain name of the C&C server 12 by attempting to recognize a signature of the malware. For example, the above-mentioned MOV and PUSH operation codes may also be used to recuperate a ciphering key hidden in the malware. Character strings in the malware may be encoded to not be easily recognizable or extractable from the binary code of the malware. An exclusive OR (XOR) operation may be made between hidden character strings and the ciphering key to uncover some malware content. The ciphering key is usually hardcoded in the binary, but may change from one malware to another.

[0049] In an embodiment, the ciphering key is automatically recuperated by searching for a predetermined operation code sequence that is indicative of a ciphering routine used in the malware. This opcode sequence may contain other operation codes besides PUSH and MOV. As a non-limiting example, Figure 7 is an illustration of a routine used by the malware MIRAI to encrypt character chains. A On Figure 7, a routine 500 defines a value 510 labelled "table_key". The table_key 510 is placed in variables k1, k2, k3 and k4 using SHIFT operation codes. A SHIFT 24 operation code 520 actually shifts the table_key 510 by 24 bits into variable k4. While SHIFT operations by 8 or 16 bits are not uncommon, the SHIFT 24 operation code 520 is an infrequently (or rarely) used operation code and provides a clue to the reverse engineering process of the location of the ciphering key. Otherwise stated, the SHIFT 24 operation code 520 is part of a signature of the malware MIRAI. Previous experience acquired from reverse engineering applied to other malwares may be put to use to identify specific operation codes as potential markers for corresponding malwares.

[0050] The variables k1, k2, k3 and k4, which include k4 that is the object of the SHIFT 24 operation code 520, are used in XOR operations 530. The application of the XOR operations 530 on the variable k4 having be the object of the SHIFT 24 operation code 520 provides a strong clue that the table key 510 is actually the ciphering key that may be used to decipher the malware and then to extract the address or the domain name of the C&C server 12. A value is read from the EAX register and shifted using a "mov eax, dword <addr>" instruction. The address <addr> is read. Inspection of an unsigned 32-bit integer (uint32) value at the address <addr> reveals the ciphering key.

[0051] Figure 8 is a sequence diagram showing detailed operations of the reverse engineering method for extracting an address or a domain name of a C&C server in accordance with an embodiment of the present technology.

[0052] On Figure 8, a sequence 600 implementing the reverse engineering method comprises a plurality of operations that may be executed in variable order, some of the operations possibly being executed concurrently, some of the operations being optional. The sequence 600 shows operations that take place in the C&C data collector 130. In an embodiment, the

C&C data collector 130 may be part of the infrastructure 100, as illustrated in the example of Figure 2. Another embodiment is also contemplated, in which the C&C data collector 130 may be a stand-alone component that receives from any entity a request for the extraction from a malware of the address or of the domain name of a C&C server 12. The following paragraphs provide a non-limiting example in which the C&C data collector 130 directly receives the malware from the software decoy 120.

[0053] A malware having been received at the infrastructure 100 (operation 420 of Figure 5A), the C&C data collector 130 verifies whether or not the malware is encrypted at operation 610. If the malware is not encrypted, the C&C data collector 130 directly reads the address or the domain name of the C&C server 12 at operation 620, for example by searching in the malware an unsigned 32-bit integer (uint32) value, or a plain null-terminated string, associated with a machine language instruction dedicated to manipulating addresses.

[0054] If the malware is encrypted, the C&C data collector 130 uses previously detected ciphering keys of known malwares at operation 630 to attempt deciphering the malware. At operation 640, the C&C data collector 130 determines whether the malware has been successfully decrypted, at operation 630, using one of the previously detected ciphering keys. If the decryption is found to be successful at operation 640, the C&C data collector 130 directly reads the address or the domain name of the C&C server 12 at operation 620.

[0055] If operation 640 reveals that the malware is still encrypted, the C&C data collector 130 performs an automatic, static analysis of a binary of the malware at operation 650. In a non-limiting embodiment, operation 650 may include one or more of sub-operations 652, 654, 656, 658 and 660. At sub-operation 652, the C&C data collector 130 locates a predetermined machine language instruction sequence in the malware. The predetermined machine language instruction sequence may comprise one or more instructions, at least one of these instructions being a rarely used instruction. At sub-operation 654, the C&C data collector 130 locates a ciphering key on which the predetermined machine language instruction sequence is applied in the malware. At sub-operation 656, the C&C data collector 130 extracts the ciphering key. At sub-operation 658, the C&C data collector 130 deciphers the malware using the ciphering key. Then at sub-operation 660, the C&C data collector 130 cates the address or the domain name of the C&C server in the deciphered malware, for example by applying an XOR operation between the binary of the malware and the ciphering key.

[0056] Figure 9 is a block diagram of a computer platform in accordance with an embodiment of the present technology. A computer platform 700, which is part of the infrastructure 100, comprises a processor 710, a memory device 720, an input device 730 and an output device 740. The memory device 720, the input device 730 and the output device 740 are all operatively connected to and controlled by the processor 710. The memory device 720 may store data related to the functions implemented in the computer platform 700. The memory device 720 may also contain a non-transitory storage medium having stored thereon instructions that the processor 710 may read and execute to implement the functions of the computer platform 700. In an implementation, the input device 730 and the output device 740

may actually consist of a combined input/output device. The input device 730 and the output device 740, or the input/output device as the case may be, allow the computer platform 700 to interconnect with other computer platforms 700 of the infrastructure 100 and/or to communicate with external entities, including without limitation the bots 18, 20, 22 and the C&C server 12. The computer platform 700 may be a general-purpose computer, a server or any system having processing capabilities. The computer platform 700 may actually be formed of a plurality of devices, for example two or more cooperating computers. The computer platform 700 may include one or more processors 710 , one or more memory devices 720, one or more input devices 730, one or more output devices 740, and/or one or more input/output devices.

[0057] Each of the game server 110, the software decoy 120, the C&C data collector 130, the client 140 and the cleaning component 150 may be implemented on the computer platform 700. In one variant, each one of the game server 110, the software decoy 120, the C&C data collector 130, the client 140 and the cleaning component 150 are implemented on a distinct and respective computer platform 700. In another variant, two (2) or more of the game server 110, the software decoy 120, the C&C data collector 130, the client 140 and the cleaning component 150 may be implemented on one computer platform 700 while the other components of the infrastructure 100 are implemented on one or more other computer platforms 700.

[0058] As mentioned hereinabove, the infrastructure 100 may include a plurality of each one of the game server 110, the software decoy 120, the C&C data collector 130, the client 140 and the cleaning component 150. In a non-limiting example, two (2) instances of the client 140 may be implemented on two (2) distinct computer platforms 700. In the same or another non-limiting example, two (2) instances of the software decoy 120 may share a common computer platform 700.

[0059] To implement the system for defending an infrastructure against a distributed denial of service (DDoS) attack, a first computer program implementing the cleaning component 150, which is adapted to discard incoming signals having at least one particular of a DDoS attack, is implemented on a computer platform 700. A second computer program configured to pose as the software decoy 120 and adapted to receive a malware intended to infect the software decoy 120 is implemented on the same or on another computer platform 700. A third computer program adapted to implement functions of C&C data collector 130 by receiving the malware from the software decoy and extracting from the malware the address or the domain name of the C&C server 12 is implemented on one of the above mentioned computer platforms or on another computer platform. A fourth computer program configured to pose as the client 140 and adapted to receive the address or the domain name of the C&C server 12 from the C&C data collector 130, use the address or the domain name of the C&C server 12 to connect to the C&C server 12, receive, a command intended by the C&C 12 server to cause the client 140 to participate in the DDoS attack, and forward particulars of the DDoS attack to the cleaning component 150 is implemented on one of the above mentioned computer platforms or on yet another computer platform.

[0060] Generally speaking and without limitation, the first computer program that implements the cleaning component 150 may implement operation 450 of Figure 5B and may further implement one or more of sub-operations 451, 452, and 453 of Figure 5B. Likewise, the second computer program that implements the software decoy 120 may implement operations 405 to 420 of Figure 5A. In the same manner, the third computer program that implements the C&C data collector 130 may implement operation 425 of Figure 5A and all operations of the sequence 600 of Figure 8. Finally, the fourth computer program that implements the client 140 may implement operations 430, 435 and 440 of Figures 5A and 5B.

[0061] Figure 10 is a graph showing a variation of a number of abuse notifications in the infrastructure before and after implementation the present technology. In more details, a graph 800 shows a variation 810 of a number of "abuse reports" over time, an abuse being reported when it is found that a component of the infrastructure 100, for example a server, is hosting a C&C server. A line 820 shows an average over time of the variation 810 prior to the introduction, in the infrastructure 100, of the workflow for defending the infrastructure 100 against DDoS attacks. After this introduction at a time 830, there was an initial surge in the detections caused by the workflow revealing previously introduced infections. Thereafter, an average over time 840 of the variation 810 of the number of abuse reports shows a very significant decrease in the number of infections.

[0062] Figure 11 is a graph showing a variation of a number of C&C servers hosted in the infrastructure before and after implementation the present technology. A curve 850 shows a number of infected servers in the infrastructure 100. Prior to the introduction of the workflow at time 830, up to 20-25 percent of the servers were being turned into C&C servers. Following the introduction of the workflow, the curve 850 shows that the number of infected servers has fallen to a range of 0-5%. Figure 12 is a graph showing a worldwide variation of a number of infections in the same timescale as in Figure 11. A curve 860 and a trend 870 of the curve 860 show that, on a worldwide basis, monthly detections of infections by infected IoT devices is continuously increasing. Comparing the results of Figures 11 and 12 shows the efficiency of the present technology.

[0063] While the above-described implementations have been described and shown with reference to particular steps performed in a particular order, it will be understood that these steps may be combined, sub-divided, or re-ordered without departing from the teachings of the present technology. At least some of the steps may be executed in parallel or in series. Accordingly, the order and grouping of the steps is not a limitation of the present technology.

[0064] It should be expressly understood that not all technical effects mentioned herein need to be enjoyed in each and every embodiment of the present technology.

[0065] Modifications and improvements to the above-described implementations of the present technology may become apparent to those skilled in the art. The foregoing description is intended to be exemplary rather than limiting. The scope of the present invention is therefore intended to be limited solely by the scope of the appended claims.

Patentkrav

1. Fremgangsmåde til at forsvare en infrastruktur (100) mod et distributed denial of service, DDoS, -angreb, omfattende:

at modtage (420), ved et softwarelokkemiddel (120) i infrastrukturen (100), en malware beregnet til at inficere softwarelokkemidlet (120);

at modtage malwaren fra softwarelokkemidlet (120) ved en kommando- og kontrol, C&C, -datasamler (130) i infrastrukturen (100); og

at uddrage (425) fra malwaren, med C&C-datasamleren (120), en adresse eller et domænenavn på en C&C-server (12);

kendetegnet ved, at fremgangsmåden yderligere omfatter:

at sende, fra C&C-datasamleren (130) til en klient (140) i infrastrukturen (100), adressen eller domænenavnet på C&C-serveren (12);

at anvende (430), med klienten (140), adressen eller domænenavnet på C&C-serveren (12) til at forbinde klienten (140) til C&C-serveren (12);

at modtage (435), ved klienten (140), en kommando beregnet af C&C-serveren (12) til at få klienten (140) til at deltage i DDoS-angrebet;

at videresende (440) oplysninger om DDoS-angrebet fra klienten (140) til en oprydningskomponent (150) i infrastrukturen (100); og

at kassere (450), i oprydningskomponenten (150), indgående signaler med mindst en af oplysningerne om DDoS-angrebet.

2. Fremgangsmåde ifølge krav 1, yderligere omfattende, forud for modtagelse (420) af malwaren ved softwarelokkemidlet (120):

at modtage (410), ved softwarelokkemidlet (120), en en udfordring beregnet til at detektere en beskyttelsesfunktion i infrastrukturen (100); og

at videresende (415) et udfordringsrespons fra softwarelokkemidlet (120).

3. Fremgangsmåde ifølge krav 1 eller 2, hvor oplysningerne om DDoS-angrebet omfatter en adresse på et tilsigtet offer for DDoS-angrebet.

4. Fremgangsmåde ifølge krav 3, yderligere omfattende at opdatere (445) en rutningstabel af infrastrukturen (100) for at forårsage rutning af indgående beskeder destineret til adressen på det tilsigtede offer i retning af oprydnings-

komponenten (150), når det tilsigtede offer er del af infrastrukturen (100).

5. Fremgangsmåde ifølge et hvilket som helst af kravene 1 til 4, hvor at kassere (450), i oprydningskomponenten (150), de indgående signaler, som har mindst en af oplysningerne om DDoS-angrebet, omfatter et element valgt fra at kassere (451) indgående signaler, som bærer falske kilde IP-adresser, at kassere (452) indgående signaler, der ikke er relateret til tidligere oprettede forbindelser, at kassere (453) indgående signaler når et antal indgående signaler, som bærer en samme kilde IP-adresse, overstiger en forudbestemt tærskel, og en kombination deraf.

6. Fremgangsmåde ifølge et hvilket som helst af kravene 1 til 5, hvor at uddrage (425) fra malwaren adressen eller domænenavnet på C&C-serveren (12) omfatter:

at verificere (610) om malwaren er krypteret;
direkte at læse (620) adressen eller domænenavnet på C&C-serveren (12), hvis malwaren ikke er krypteret;
hvis malwaren er krypteret, at anvende (630) en eller flere tidligere detekterede krypteringsnøgler fra kendte malware til at afkode malwaren;
at verificere (640) om malwaren stadig er krypteret efter anvendelse af den ene eller flere tidligere detekterede krypteringsnøgler; og
hvis malwaren stadig er krypteret, at udføre (650) en automatisk, statisk analyse af en binær af malwaren.

7. Fremgangsmåde ifølge krav 6, hvor at udføre (650) den automatiske, statiske analyse af det binære af malwaren omfatter:

at lokalisere (652) en forudbestemt maskinsprogsinstruktionssekvens i malwaren;
at lokalisere (654) en krypteringsnøgle på hvilken den forudbestemte maskinsprogsinstruktionssekvens anvendes i malwaren;
at uddrage (656) krypteringsnøglen fra malwaren;
at afkode (658) malwaren under anvendelse af krypteringsnøglen; og
at lokalisere (660) adressen eller domænenavnet på C&C-serveren (12) i den afkodede malware.

8. Fremgangsmåde ifølge et hvilket som helst af kravene 1 til 7, yderligere omfattende:

at bestemme (445) om en downloader af malwaren eller C&C-serveren (12) er hosted i infrastrukturen (100) ved at verificere adressen eller domænenavnet på downloaderen af malwaren eller C&C-serveren (12); og hvis downloaderen af malwaren eller C&C-serveren (12) er hosted i en kompromitteret komponent i infrastrukturen (100), at udføre (445) en handling valgt fra at advare en kunde, som har indhold hosted i den kompromitterede komponent i infrastrukturen, at anbringe den kompromitterede komponent i infrastrukturen i karantæne, og en kombination deraf.

9. System til at forsvare en infrastruktur (100) mod et distributed denial of service, DDoS, -angreb, omfattende:

et softwarelokkemiddel (120) for infrastrukturen, indrettet til at modtage en malware beregnet til at inficere softwarelokkemidlet (120); og

en kommando- og kontrol, C&C, -datasamler (130 for infrastrukturen,) indrettet til:

at modtage malwaren fra softwarelokkemidlet (120); og

at uddrage fra malwaren en adresse eller et domænenavn på en C&C-server (12);

kendetegnet ved, at systemet yderligere omfatter:

en oprydningskomponent (150) for infrastrukturen; og

en klient (140) for infrastrukturen, indrettet til:

at modtage adressen eller domænenavnet på C&C-serveren (12) fra C&C-datasamleren (130);

at anvende adressen eller domænenavnet på C&C-serveren (12) til at forbinde til C&C-serveren (12),

at modtage, en kommando tilsigtet af C&C-serveren (12) til at forårsage klienten (140) til at deltage i DDoS-angrebet, og

at videresende oplysninger om DDoS-angrebet til oprydningskomponenten (150);

hvor oprydningskomponenten er indrettet til at kassere indgående signaler, som har mindst en af oplysingerne om DDoS-angrebet.

10. System ifølge krav 9, hvor C&C-datasamleren (130) er yderligere indrettet til at uddrage fra malwaren adressen eller domænenavnet på C&C-serveren (12) ved:

direkte at læse adressen eller domænenavnet på C&C-serveren (12), hvis malwaren ikke er krypteret;

at anvende en eller flere tidligere detekterede krypteringsnøgler fra kendte malware til at afkode malwareserveren, hvis malwaren er krypteret; og

hvis malwaren stadig er krypteret efter anvendelse af den ene eller flere tidligere detekterede krypteringsnøgler, at udføre en automatisk, statisk analyse af en binær af malwaren.

11. System ifølge krav 10, hvor C&C-datasamleren (130) er yderligere indrettet til at udføre den automatiske, statiske analyse af den binære af malwaren ved:

at lokalisere en forudbestemt maskinsprogsinstruktionssekvens i malwaren;

at lokalisere en krypteringsnøgle på hvilken den forudbestemte maskinsprogsinstruktionssekvens anvendes i malwaren;

at uddrage krypteringsnøglen fra malwaren;

at afkode malwaren under anvendelse af krypteringsnøglen; og

at lokalisere adressen eller domænenavnet på C&C-serveren (12) i den afkodede malware.

12. System ifølge krav 11, hvor den forudbestemte maskinsprogsinstruktions-sekvens omfatter en eller flere instruktioner inklusive mindst en sjældent anvendt instruktion.

13. System ifølge et hvilket som helst af kravene 9 til 12, hvor software-lokkemidlet (120) ikke installerer malwaren.

14. System ifølge et hvilket som helst af kravene 9 til 13, hvor systemet er yderligere indrettet til:

at bestemme om en downloader af malwaren eller C&C-serveren (12) er hosted i infrastrukturen (100) ved at verificere adressen eller domænenavnet på downloaderen af malwaren eller C&C-serveren (12); og

hvis downloaderen af malwaren eller C&C-serveren (12) er hosted i en kompromitteret komponent i infrastrukturen (100), at forårsage infrastrukturen (100) til at udføre en handling valgt fra at advare en kunde, som har indhold hosted i den kompromitterede komponent i infrastrukturen, (100) at anbringe den kompromitterede komponent fra infrastrukturen (100) i karantæne, og en kombination deraf.

DRAWINGS

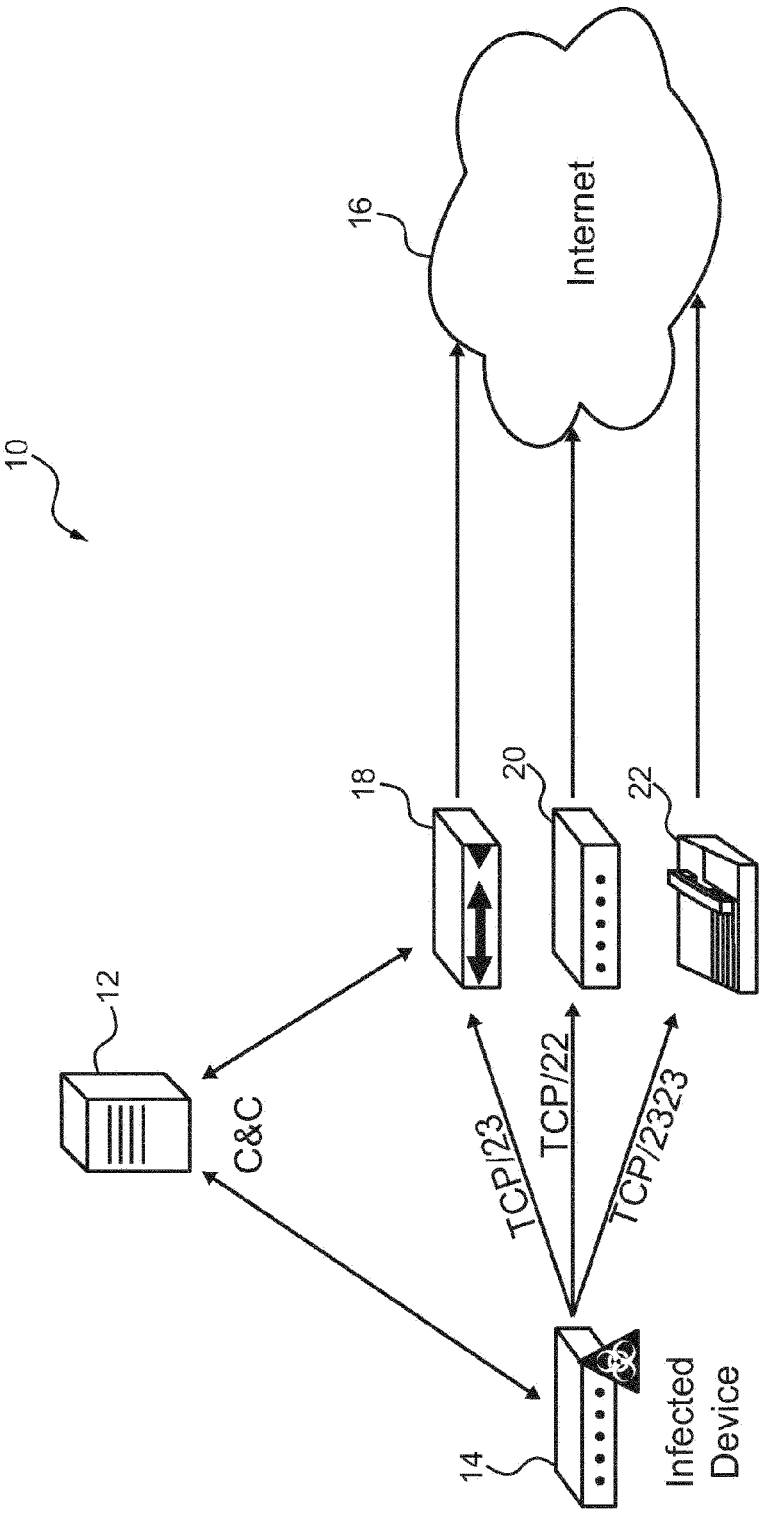

Figure 1 (Prior Art)

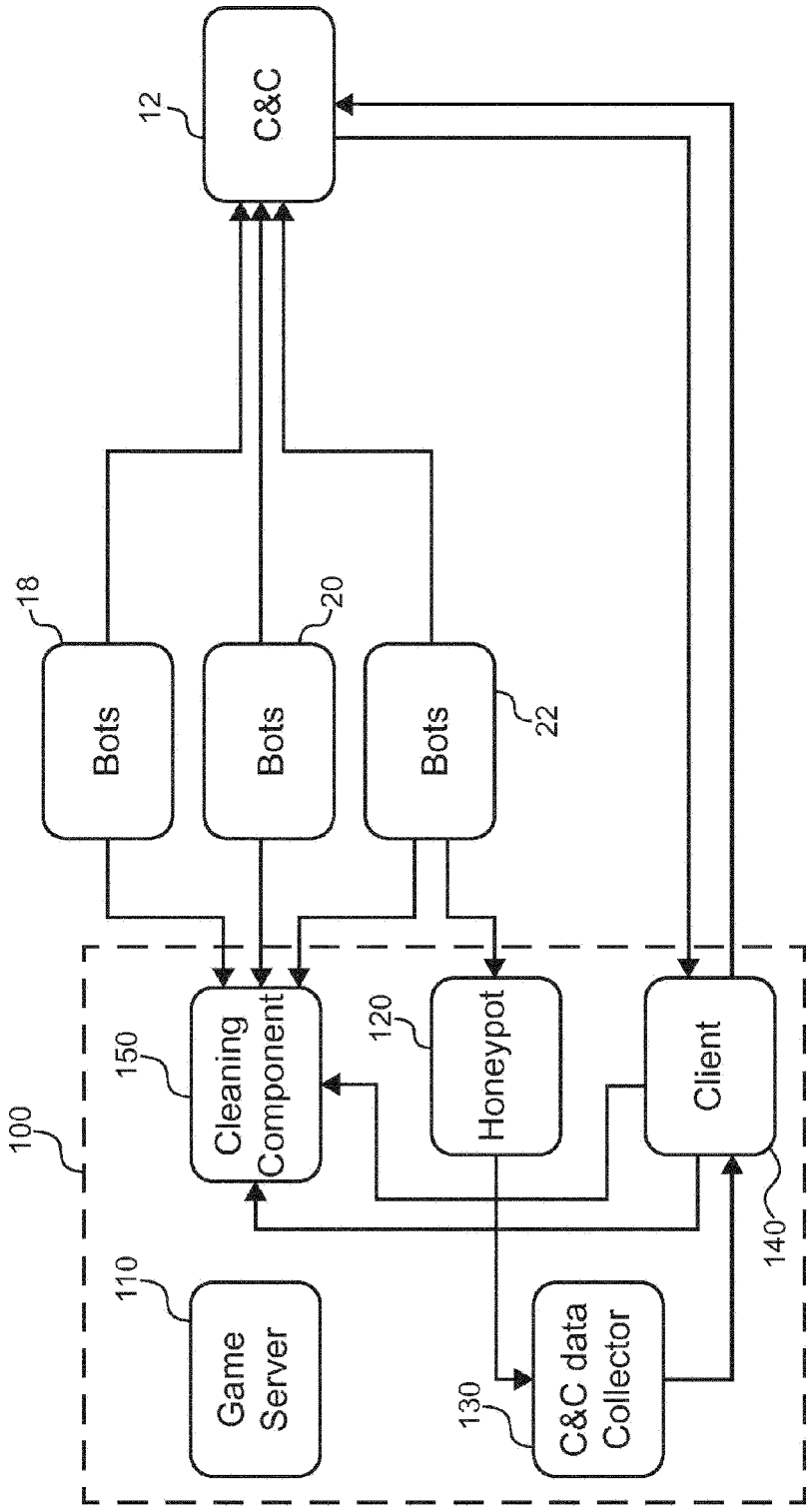
C&C
12
Bots
18
Bots
20
Bots
22
100
Cleaning Component
150
Honeypot
120
Client
140
Game Server
110
C&C data Collector
130

Figure 2

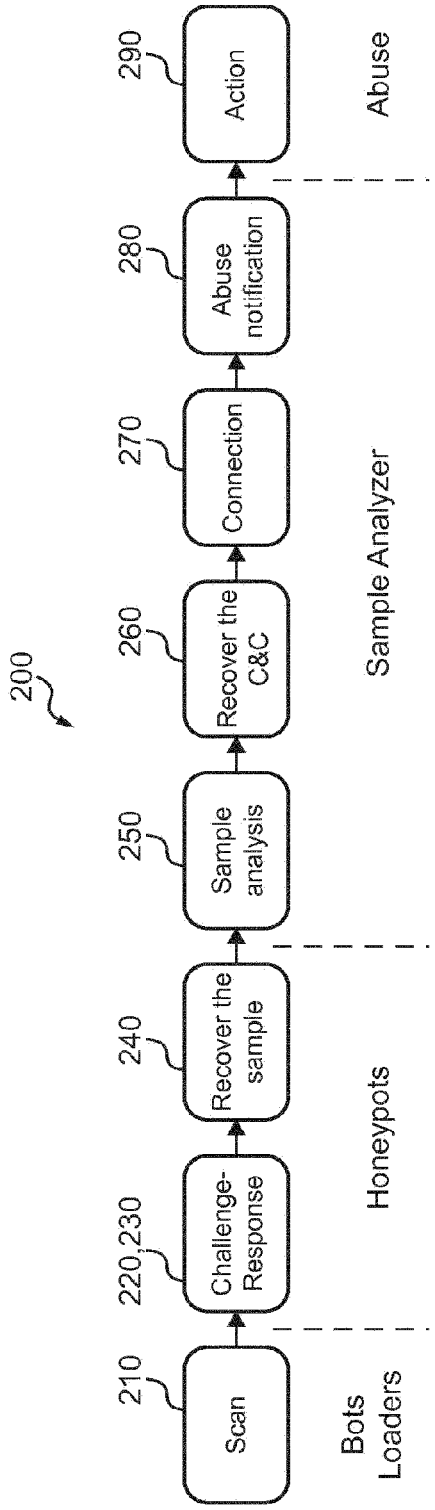

Figure 3

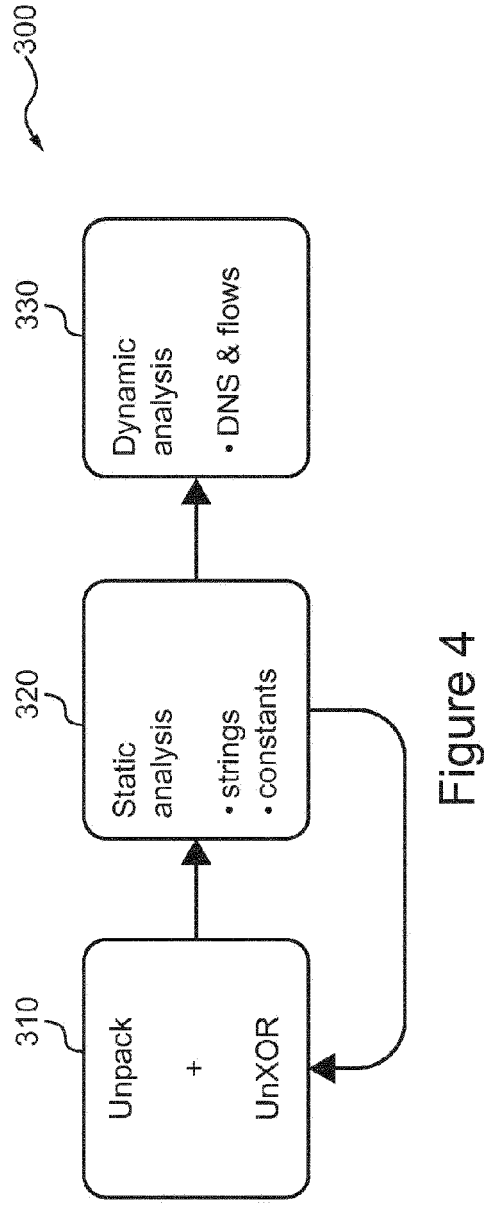

Figure 4

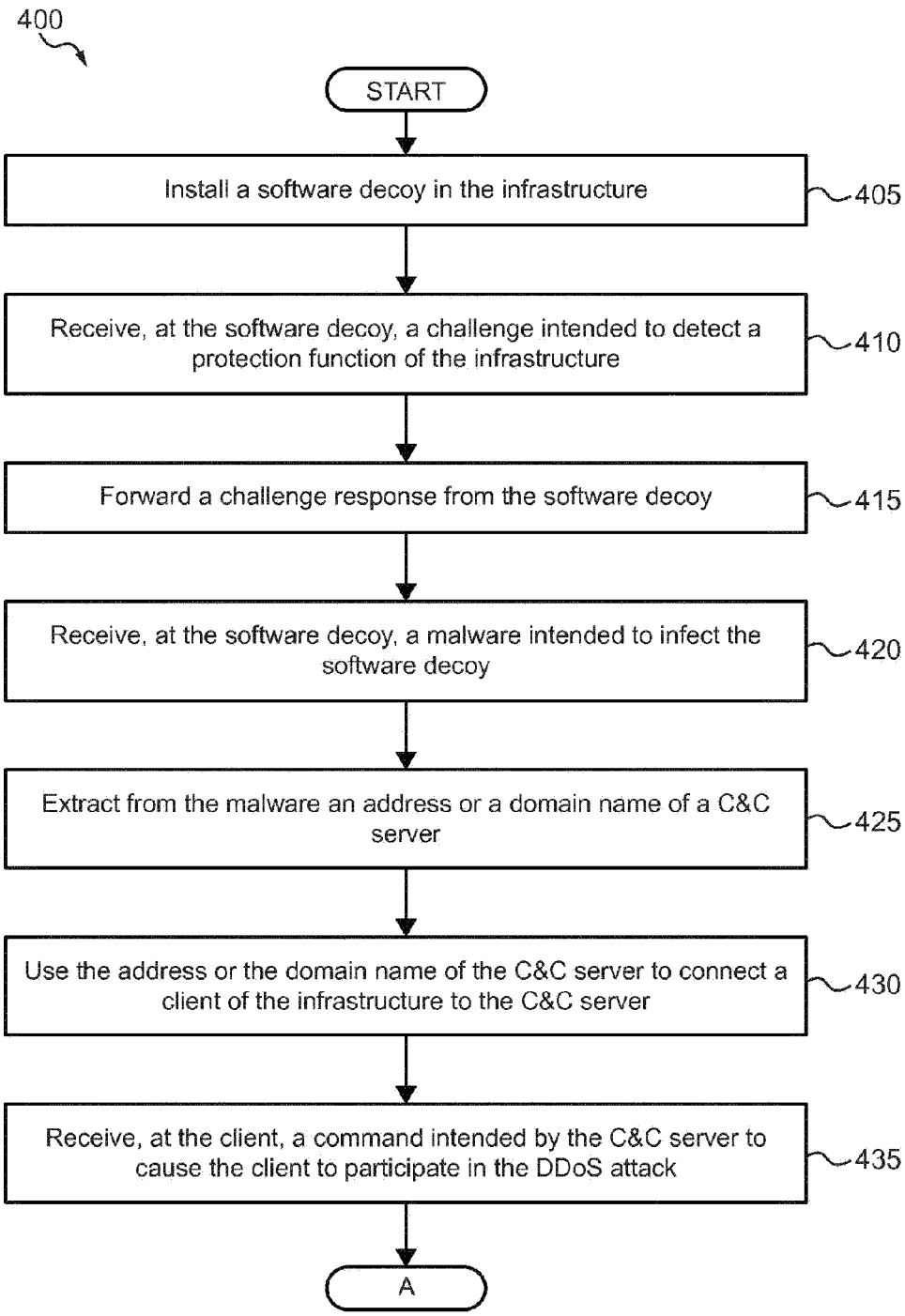
400
START
Install a software decoy in the infrastructure
405
Receive, at the software decoy, a challenge intended to detect a protection function of the infrastructure
410
Forward a challenge response from the software decoy
415
Receive, at the software decoy, a malware intended to infect the software decoy
420
Extract from the malware an address or a domain name of a C&C server
425
Use the address or the domain name of the C&C server to connect a client of the infrastructure to the C&C server
430
Receive, at the client, a command intended by the C&C server to cause the client to participate in the DDoS attack
435
A

Figure 5A

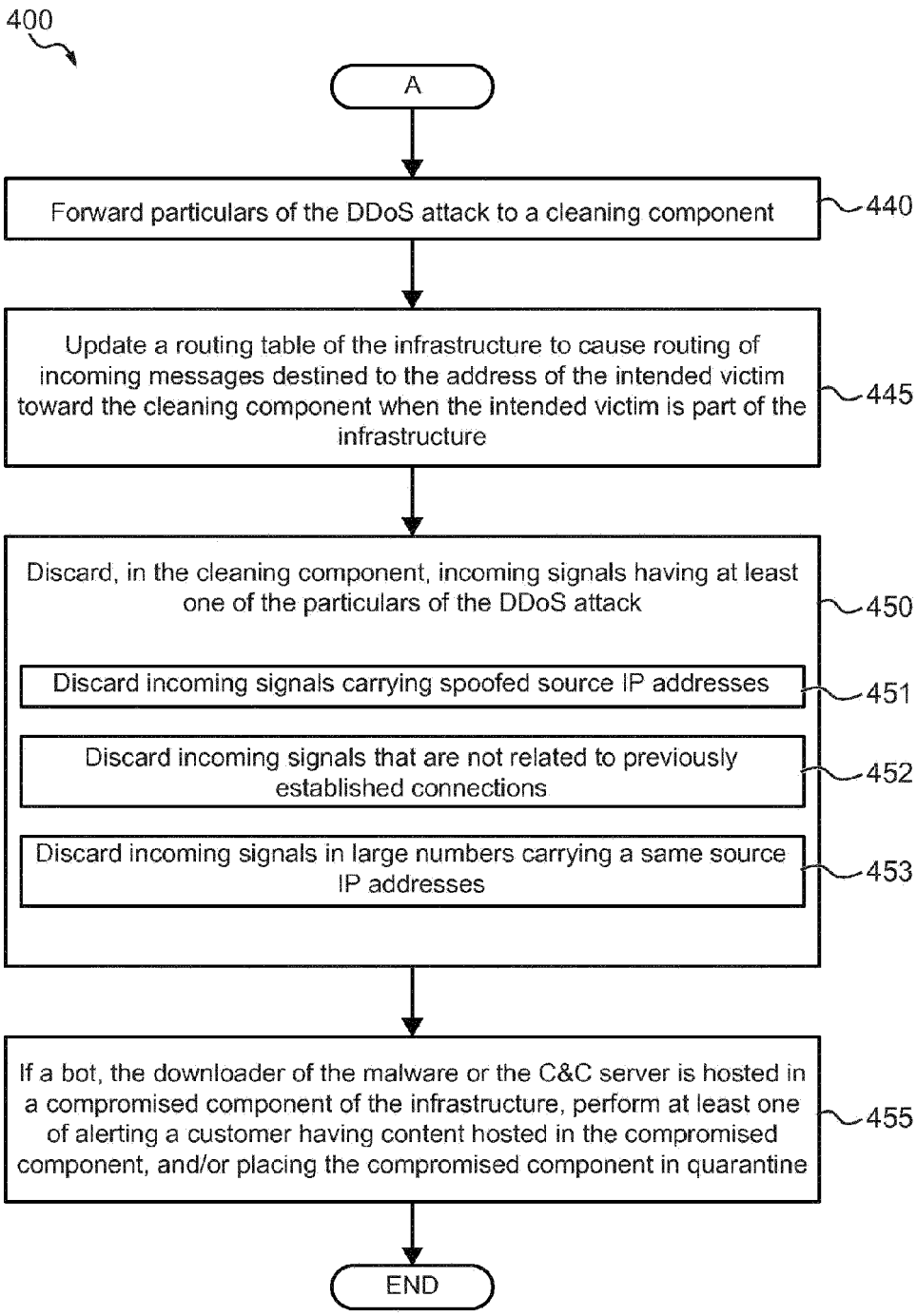
400
A
Forward particulars of the DDoS attack to a cleaning component
440
Update a routing table of the infrastructure to cause routing of incoming messages destined to the address of the intended victim toward the cleaning component when the intended victim is part of the infrastructure
445
Discard, in the cleaning component, incoming signals having at least one of the particulars of the DDoS attack
450
Discard incoming signals carrying spoofed source IP addresses
451
Discard incoming signals that are not related to previously established connections
452
Discard incoming signals in large numbers carrying a same source IP addresses
453
If a bot, the downloader of the malware or the C&C server is hosted in a compromised component of the infrastructure, perform at least one of alerting a customer having content hosted in the compromised component, and/or placing the compromised component in quarantine
455
END

Figure 5B

```
Jun 26 16:20:27 - !* GTFONIGG!* GTFOFAG
Jun 26 16:20:27 - !* GTFODUP
Jun 26 16:25:19 - !* STD .XX.XXX.XXX.70 443 200
Jun 26 16:34:15 - !* STD .XX.XXX.XXX.104 5355 200
Jun 26 16:34:37 - !* UOP .XX.XXX.XXX.104 433 200 32 0 1
Jun 26 16:34:48 - !* STOPATTK
Jun 26 16:34:57 - !* GTFONIGG!* GTFOFAG
Jun 26 16:34:57 - !* GTFODUP
Jun 26 16:35:26 - !* STD .XX.XXX.XXX.101 443 100
Jun 26 16:37:09 - !* STOPATTK
Jun 26 16:37:44 - !* STD .XX.XXX.XXX.100 5355 200
Jun 26 16:37:51 - !* STOPATTK
Jun 26 16:42:26 - !* STD .XX.XXX.XXX.67 443 100
Jun 26 16:43:09 - !* STD .XX.XXX.XXX.67 22 200
Jun 26 16:45:00 - !* STD .XX.XXX.XXX.42 23 200
Jun 26 16:47:40 - !* STD .XX.XXX.XXX.42 443 200
Jun 26 16:51:18 - !* STD .XX.XXX.XXX.214 443 200
Jun 26 16:52:53 - !* STD .XX.XXX.XXX.74 5355 200
Jun 26 16:53:08 - !* STD .XX.XXX.XXX.74 21 200
Jun 26 16:54:30 - !* SCANNER ON
Jun 26 16:54:30 - !* FATCOCK
Jun 26 16:56:43 - !* STD .XX.XXX.XXX.74 21 200
Jun 26 16:56:50 - !* STD .XX.XXX.XXX.74 443 200
Jun 26 16:56:57 - !* STOPATTK
Jun 26 17:23:13 - !* GTFONIGG!* GTFOFAG
```

Figure 6

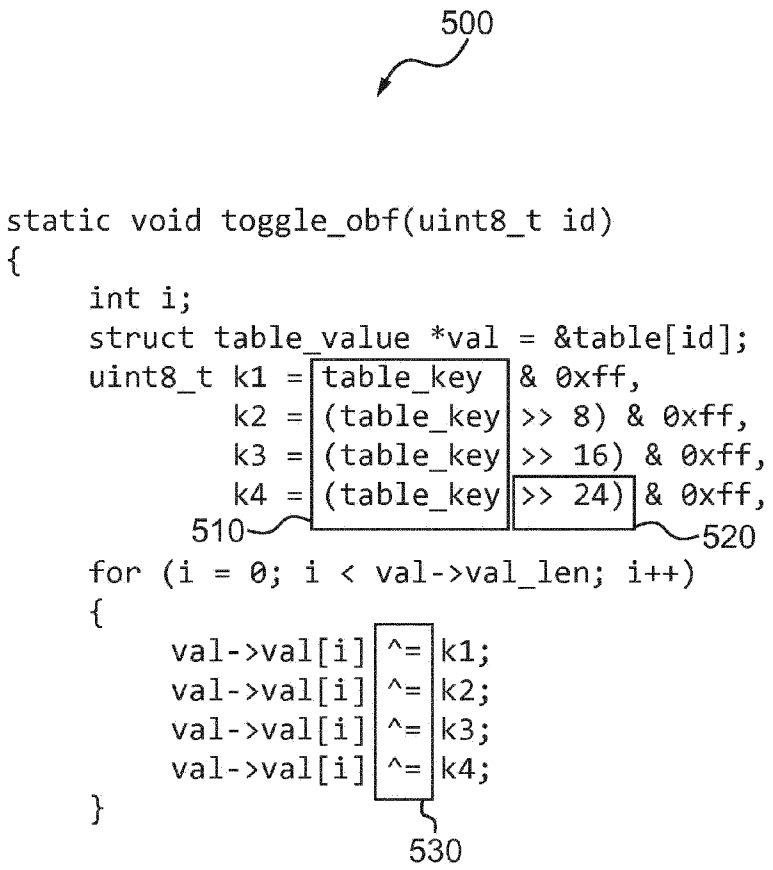
500
static void toggle_obf(uint8_t id)
{
int i;
struct table_value *val = &table[id];
uint8_t k1 = table_key & 0xff,
k2 = (table_key >> 8) & 0xff,
k3 = (table_key >> 16) & 0xff,
k4 = (table_key >> 24) & 0xff,
510
520
for (i = 0; i < val->val_len; i++)
{
val->val[i] ^= k1;
val->val[i] ^= k2;
val->val[i] ^= k3;
val->val[i] ^= k4;
}
530
#ifdef DEBUG
val->locked = !val->locked;
#endif
}

Figure 7

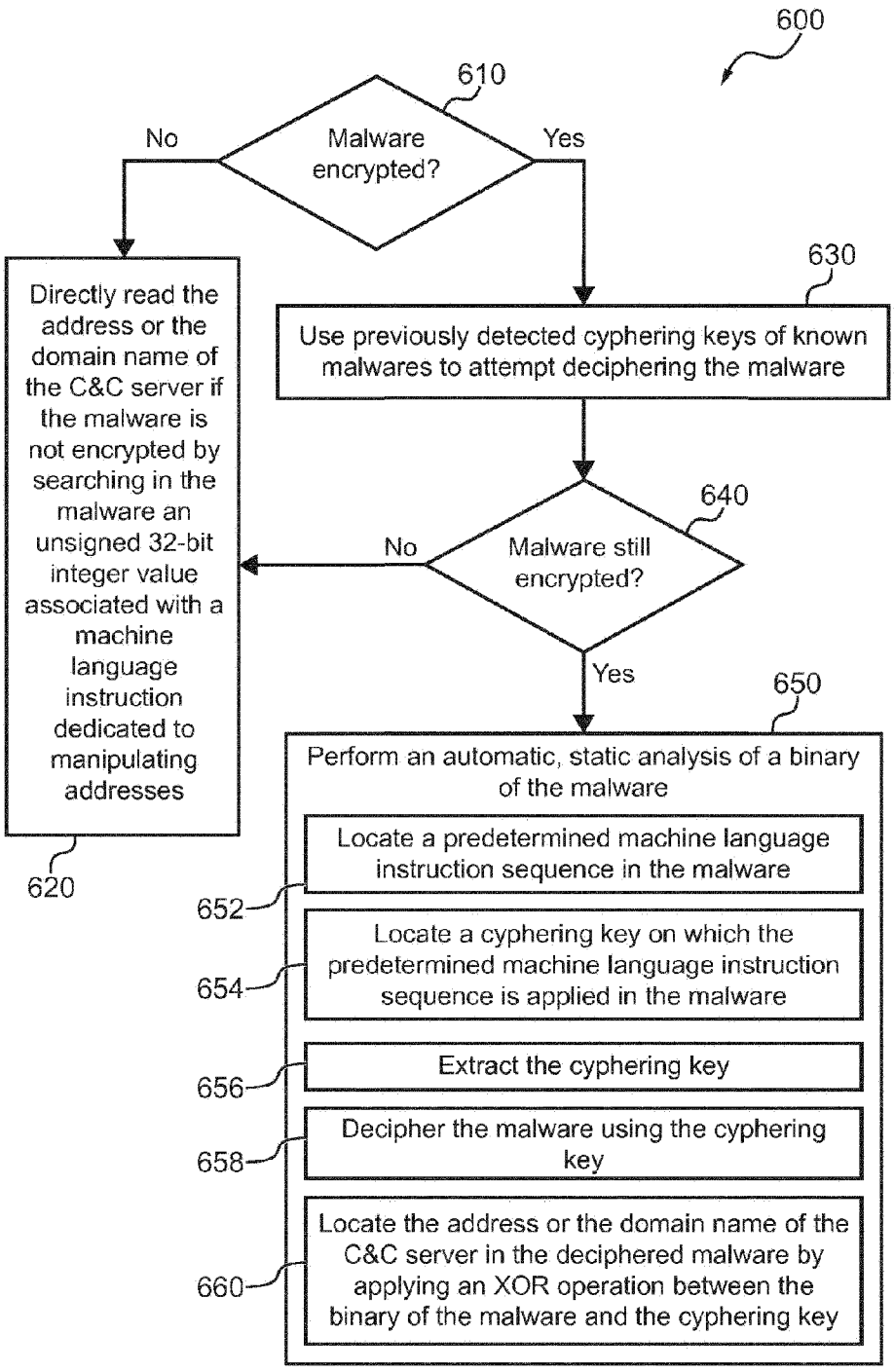
600
610
Malware encrypted?
No
Yes
Directly read the address or the domain name of the C&C server if the malware is not encrypted by searching in the malware an unsigned 32-bit integer value associated with a machine language instruction dedicated to manipulating addresses
620
630
Use previously detected cyphering keys of known malwares to attempt deciphering the malware
640
Malware still encrypted?
No
Yes
650
Perform an automatic, static analysis of a binary of the malware
652
Locate a predetermined machine language instruction sequence in the malware
654
Locate a cyphering key on which the predetermined machine language instruction sequence is applied in the malware
656
Extract the cyphering key
658
Decipher the malware using the cyphering key
660
Locate the address or the domain name of the C&C server in the deciphered malware by applying an XOR operation between the binary of the malware and the cyphering key

Figure 8

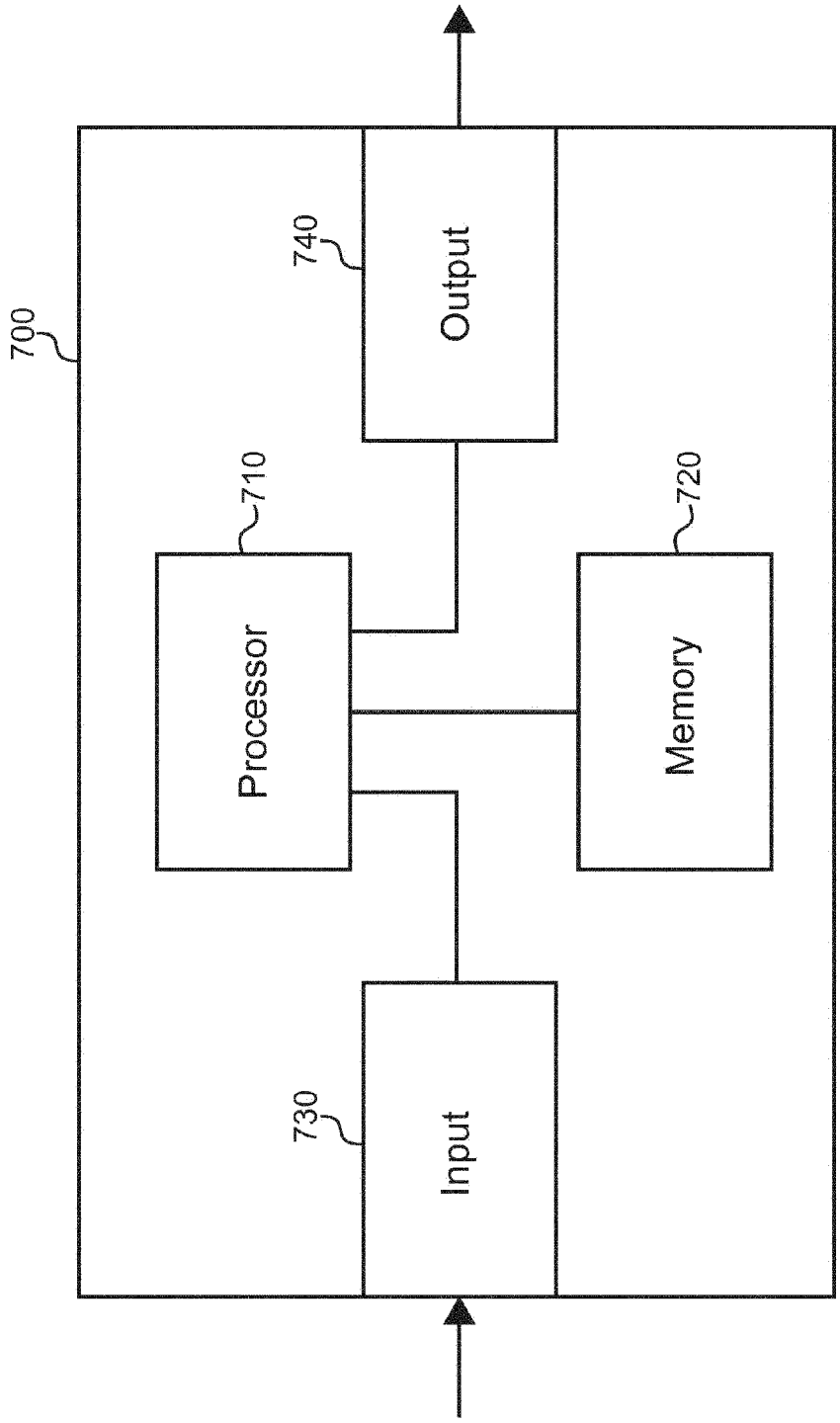

Figure 9

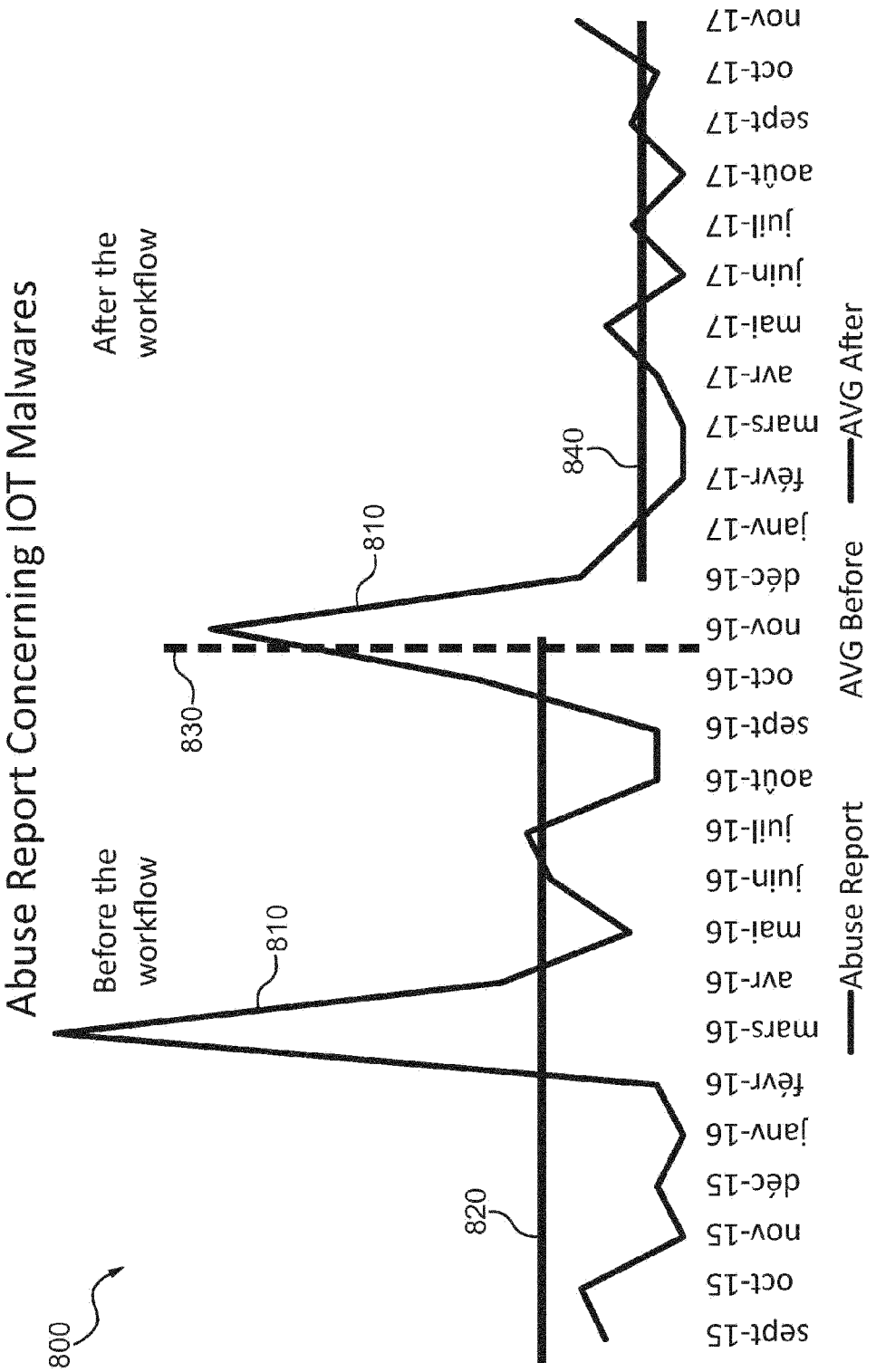
Abuse Report Concerning IOT Malwares
After the workflow
Before the workflow
800
810
810
820
830
840
sept-15
oct-15
nov-15
déc-15
janv-16
févr-16
mars-16
avr-16
mai-16
juin-16
juil-16
août-16
sept-16
oct-16
nov-16
déc-16
janv-17
févr-17
mars-17
avr-17
mai-17
juin-17
juil-17
août-17
sept-17
oct-17
nov-17
Abuse Report
AVG Before
AVG After

Figure 10

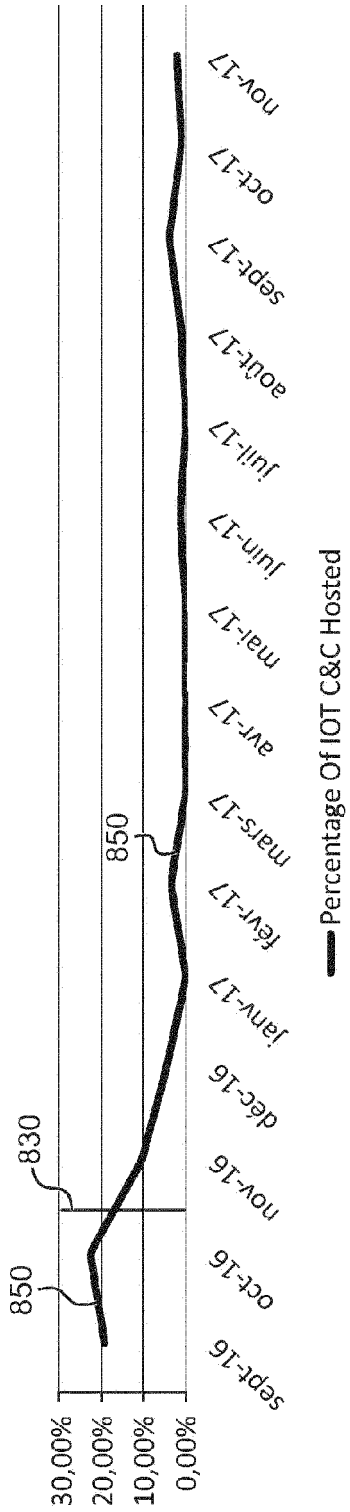

Figure 11

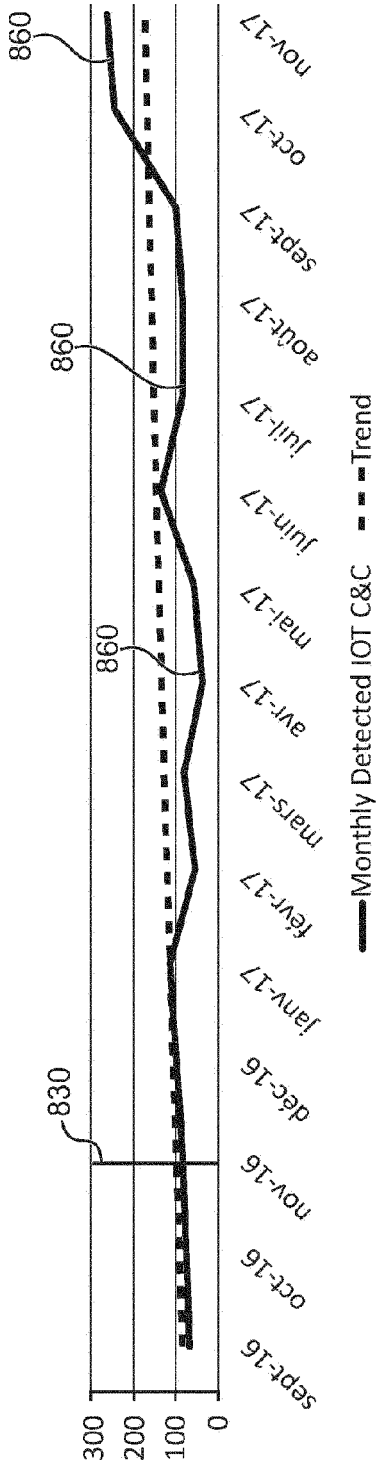

Figure 12