

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
12 March 2009 (12.03.2009)

PCT

(10) International Publication Number
WO 2009/032379 A1

(51) International Patent Classification:
G06F 12/14 (2006.01)

(74) Agents: **POH, Philip, R.** et al.; Byrne Poh LLP, 11 Broadway, Suite 865, New York, NY 10004 (US).

(21) International Application Number:
PCT/US2008/066623

(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, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(22) International Filing Date: 12 June 2008 (12.06.2008)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/934,307 12 June 2007 (12.06.2007) US
61/044,376 11 April 2008 (11.04.2008) US

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

(71) Applicant (for all designated States except US): **THE TRUSTEES OF COLUMBIA UNIVERSITY IN THE CITY OF NEW YORK** [US/US]; 412 Low Memorial Library, 535 West 116th Street, New York, NY 10027 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **STOLFO, Salvatore, J.** [US/US]; 80 Kenilworth Road, Ridgewood, NJ 07450 (US). **KEROMYTIS, Angelos, D.** [GR/US]; 423 West 120th Street #98, New York, NY 10027 (US).

Published:

- with international search report
- before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments

(54) Title: METHODS AND SYSTEMS FOR PROVIDING TRAP-BASED DEFENSES

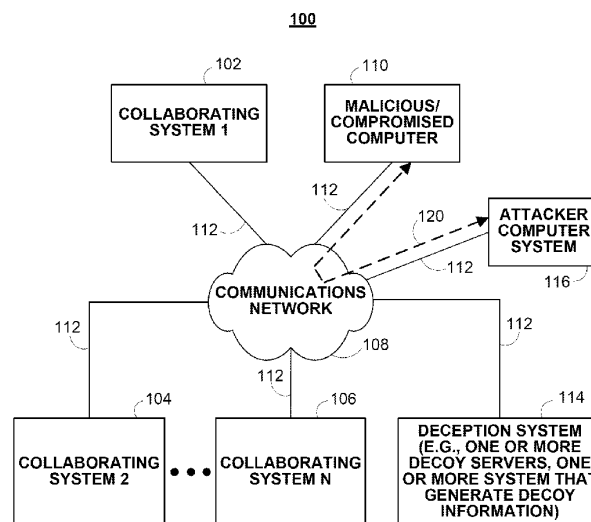


FIG. 1

(57) Abstract: Methods and systems for providing trap-based defenses are provided. In accordance with some embodiments, methods for providing trap-based defenses are provided, the method comprising: generating decoy information; embedding a beacon into the decoy information; and inserting the decoy information with the embedded beacon into a computing environment, wherein the embedded beacon provides an indication that the decoy information has been accessed by an attacker.

WO 2009/032379 A1

METHODS AND SYSTEMS FOR PROVIDING TRAP-BASED DEFENSES

Cross Reference to Related Application

[0001] This application claims the benefit of United States Provisional Patent Application No. 60/934, 307, filed June 12, 2007 and United States Provisional Patent Application No. 61/044,376, filed April 11, 2008, which are hereby incorporated by reference herein in their entireties.

[0002] This application is also related to International Application No. PCT/US2007/012811, filed May 31, 2007, which is hereby incorporated by reference herein in its entirety.

Statement Regarding Federally Sponsored Research Or Development

[0003] The invention was made with government support under Grant No. DARTC-5-36423-5740 awarded by the U.S. Department of Homeland Security through the Institute for Information Infrastructure Protection (I3P). The government has certain rights in the invention.

Technical Field

[0004] The disclosed subject matter relates to methods and systems for providing trap-based defenses.

Background

[0005] Computer viruses, worms, trojans, hackers, rootkits, spyware, key recovery attacks, denial-of-service attacks, malicious software (or malware), probes, etc. are a constant menace to all users of computers connected to public computer networks (such as the Internet) and/or private networks (such as corporate computer networks). Because of these threats, many computers are protected by antivirus software and firewalls. However, these preventative measures are not always adequate. For example, documents can be embedded with malware (e.g., network sniffers, keystroke loggers, etc.) and inserted into a network or a system through a universal serial bus (USB) drive, a compact disk (CD), or downloaded from a reputable source, thereby bypassing preventative measures like firewalls and packet filters.

[0006] In addition, the removal of malware is not always possible. In many situations, malicious code can sometimes be identified, but because the infected system performs a mission-critical operation, the infection is allowed to continue since the system cannot be stopped due to the operational need to keep the system running or the requirement to keep the data intact. This has made filtering-based prevention mechanisms an ineffective and an insufficient defense. In failing to stop the malware, malware can reach systems in a network and cause serious damage, particularly if the malware is left undetected for long periods of time.

[0007] Using malware or other threats, attackers can snoop or eavesdrop on a computer or a network, download and exfiltrate data, steal assets and information, destroy critical assets and information, and/or modify information. Moreover, it should also be noted that these filtering-based prevention mechanisms are equally ineffective against inside attackers (e.g., human agents or their system, software proxies, etc.). Similar to attackers that gain access to a computer through malware, inside attackers can perform unauthorized activities, such as exfiltrate data, steal assets, destroy critical information, and/or modify information. This has become one of the most serious threats encountered in modern organizations.

[0008] There is therefore a need in the art for approaches that provide trap-based defenses for detecting attacks. Accordingly, it is desirable to provide methods and systems that overcome these and other deficiencies of the prior art.

Summary

[0009] Methods and systems for providing trap-based defenses are provided. In accordance with some embodiments, methods for providing trap-based defenses are provided, the method comprising: generating decoy information; embedding a beacon into the decoy information; and inserting the decoy information with the embedded beacon into a computing environment, wherein the embedded beacon provides an indication that the decoy information has been accessed by an attacker.

Brief Description of the Drawings

[0010] FIG. 1 is a diagram of a system suitable for implementing an application that inserts decoy information with embedded beacons in accordance with some embodiments.

[0011] FIG. 2 is a diagram showing an original document and a decoy document with one or more embedded beacons in accordance with some embodiments.

[0012] FIG. 3 is a diagram showing one example of generating and inserting decoy information into an operating environment in accordance with some embodiments.

[0013] FIG. 4 is a diagram showing examples of actual information (e.g., network traffic) in an operating environment in accordance with some embodiments.

[0014] FIG. 5 is a diagram showing examples of decoy information (e.g., decoy network traffic) generated using actual information and inserted into an operating environment in accordance with some embodiments.

[0015] FIGS. 6-7 are diagrams showing an example of an interface for generating decoy information in accordance with some embodiments.

[0016] FIG. 8 is a diagram showing examples of generated decoy media (e.g., decoy documents) in accordance with some embodiments.

[0017] FIG. 9 is a diagram showing an embedded beacon in accordance with some embodiments.

[0018] FIG. 10 is a diagram showing the connection opened to an external website by an embedded beacon in accordance with some embodiments.

[0019] FIG. 11 is a diagram showing an example of a website that collects beacon signals in accordance with some embodiments.

[0020] FIG. 12 is a diagram showing one example of receiving signals from a beacon embedded in decoy information and removing malware in accordance with some embodiments.

[0021] FIG. 13 is a diagram showing one example of transmitting notifications and/or recommendations in response to receiving signals from an embedded beacon in accordance with some embodiments.

Detailed Description

[0022] In accordance with various embodiments, mechanisms for providing trap-based defenses are provided. In some embodiments, systems and methods are provided that implement proactive traps based on counter-intelligence. Traps using decoy information (sometimes referred to herein as "bait information," "bait traffic," or "decoy media") can be set up to attract and/or confuse attackers (both inside and outside attackers) and/or malware. For example, large amounts of decoy information can be inserted into the network flows and large amount of decoy documents can be stored in computing systems to lure potential attackers. Decoy information can be used to reduce the level of system knowledge of an attacker, entice the attacker to perform actions that reveal their presence and/or identities, and uncover and track the unauthorized activities of the attacker. In some embodiments, the decoy information can be associated and/or embedded with one or more beacons, where the beacons transmit signals to indicate that the decoy information has been accessed, transmitted, opened, executed, and/or misused. Alternatively, the use of the decoy information with the embedded beacon can indicate that the decoy information has been exfiltrated. Beacon signals can also include information sufficient to identify and/or trace the attacker and/or malware.

[0023] It should be noted that, while preventive defense mechanisms attempt to inhibit malware from infiltrating into a network, trap-based defenses are directed towards deceiving and detecting attackers and/or malware that have succeeded in infiltrating the network.

[0024] It should also be noted that, in some embodiments, decoy information can be automatically generated and/or deployed. In some embodiments, when the decoy information has been deployed, decoy information and the embedded beacon can self-activate when accessed or otherwise misused by an attacker or malware. In response, the self-activated decoy information can identify and monitor the trail of the attacker or the malware.

[0025] It should further be noted that, in some embodiments, decoy information can be difficult to distinguish from actual information used in the system. For example, decoy information can be generated to appear realistic and indistinguishable from actual information used in the system. If the actual information is in the English language, the decoy information is generated in the

English language and the decoy information looks and sounds like properly written or spoken English. In some embodiments, the decoy information can be acquired from human generated sources, such as web blogs and archived emails. In another example, to entice a sophisticated and knowledgeable attacker, the decoy information can be a login (e.g., an email login, a system login, a network login, a website username) that appears and functions like an actual login such that it is capable of trapping the attacker (e.g., a user with malicious intent, a misbehaving system administrator or a network security staff member, etc.). In another example, decoy information can appear to contain believable, sensitive personal information and seemingly valuable information. In general, the decoy information can be believable, variable (e.g., changing such that attackers do not identify decoy information), enticing (e.g., decoy information with particular keywords), and conspicuous (e.g., located in particular folders).

[0026] These mechanisms can be used in a variety of applications. For example, a host agent (e.g., an ActiveX control, a Javascript control, etc.) can insert decoy information with an embedded beacon among data in Microsoft Outlook (e.g., in the address book, in the notes section, etc.). In another example, the accessing or misuse of decoy information can provide a detection mechanism for attacks and, in response to accessing or misusing decoy information, the embedded beacon can transmit a signal to an application (e.g., a monitoring application, a parsing application, etc.) that identifies the location of the attacker.

[0027] FIG. 1 illustrates one embodiment of a system 100 in which the trap-based defense can be implemented. As shown, system 100 includes multiple collaborating computer systems 102, 104, and 106, a communication network 108, a malicious/compromised computer 110, communication links 112, a deception system 114, and an attacker computer system 116.

[0028] Collaborating systems 102, 104, and 106 can be systems owned, operated, and/or used by universities, businesses, governments, non-profit organizations, families, individuals, and/or any other suitable person and/or entity. Collaborating systems 102, 104, and 106 can include any number of user computers, servers, firewalls, routers, switches, gateways, wireless networks, wired networks, intrusion detection systems, and any other suitable devices. Collaborating systems 102, 104, and 106 can include one or more processors, such as a general-purpose

computer, a special-purpose computer, a digital processing device, a server, a workstation, and/or various other suitable devices. Collaborating systems 102, 104, and 106 can run programs, such as operating systems (OS), software applications, a library of functions and/or procedures, background daemon processes, and/or various other suitable programs. In some embodiments, collaborating systems 102, 104, and 106 can support one or more virtual machines. Any number (including only one) of collaborating systems 102, 104, and 106 can be present in system 100, and collaborating systems 102, 104, and 106 can be identical or different.

[0029] Communication network 108 can be any suitable network for facilitating communication among computers, servers, etc. For example, communication network 108 can include private computer networks, public computer networks (such as the Internet), telephone communication systems, cable television systems, satellite communication systems, wireless communication systems, any other suitable networks or systems, and/or any combination of such networks and/or systems.

[0030] Malicious/compromised computer 110 can be any computer, server, or other suitable device for launching a computer threat, such as a virus, worm, trojan, rootkit, spyware, key recovery attack, denial-of-service attack, malware, probe, etc. The owner of malicious/compromised computer 110 can be any university, business, government, non-profit organization, family, individual, and/or any other suitable person and/or entity. The owner of malicious/compromised computer 110 may not be aware of what operations malicious/compromised computer 110 is perform or may not be in control of malicious/compromised computer 110. Malicious/compromised computer 110 can be acting under the control of another computer (e.g., attacking computer system 116) or autonomously based upon a previous computer attack which infected computer 110 with a virus, worm, trojan, spyware, malware, probe, etc. For example, some malware can passively collect information that passes through malicious/compromised computer 110. In another example, some malware can take advantage of trusted relationships between malicious/compromised computer 110 and other systems 102, 104, and 106 to expand network access by infecting other systems. In yet another example, some malware can communicate with attacking computer system 116 through an exfiltration channel 120 to transmit confidential information (e.g., IP addresses, passwords, credit card numbers, etc.).

[0031] It should be noted that malicious code can be injected into an object that appears as an icon in a document. In response to manually selecting the icon, the malicious code can launch an attack against a third-party vulnerable application. Malicious code can also be embedded in a document, where the malicious code does not execute automatically. Rather, the malicious code lies dormant in the file store of the environment awaiting a future attack that extracts the hidden malicious code.

[0032] Some malware can passively collect information that passes through malicious/compromised computer 110. Some malware can take advantage of trusted relationships between malicious/compromised system 110 and one or more of collaborating systems 102, 104, and 106 to expand network access by infecting other systems. Some malware can communicate with attacking computer system 116 through an exfiltration channel 120 to transmit, for example, confidential information.

[0033] Alternatively, in some embodiments, malicious/compromised computer 110 and/or attacking computer system 116 can be operated by an individual or organization with nefarious intent. For example, with the use of malicious code and/or exfiltration channel 120, a user of malicious/compromised computer 110 or a user of attacking computer system 116 can perform unauthorized activities (e.g., exfiltrate data without the use of channel 120, steal information from one of the collaborating systems 102, 104, and 106), etc. It should be noted that any number of malicious/compromised computers 110 and attacking computer systems 116 can be present in system 100, but only one is shown in FIG. 1 to avoid overcomplicating the drawing.

[0034] Referring back to FIG. 1, communication links 112 can be any suitable mechanism for connecting collaborating systems 102, 104, 106, malicious/compromised computer 110, deception system 114, and attacking computer system 116 to communication network 108. Links 112 can be any suitable wired or wireless communication link, such as a T1 or T3 connection, a cable modem connection, a digital subscriber line connection, a Wi-Fi or 802.11(a), (b), (g), or (n) connection, a dial-up connection, and/or any other suitable communication link. Alternatively, communication links 112 can be omitted from system 100 when appropriate, in which case systems 102, 104, and/or 106, computer 110, and/or deception system 114 can be connected directly to communication network 108.

[0035] Deception system 114 can be any computer, server, or other suitable device for modeling, generating, inserting and/or distributing decoy information into system 100. Similar to collaborating systems 102, 104, and 106, deception system 114 can run programs, such as operating systems (OS), software applications, a library of functions and/or procedures, background daemon processes, and/or various other suitable programs. In some embodiments, deception system 114 can support one or more virtual machines.

[0036] For example, deception system 114 can be a designated server or a dedicated workstation that analyzes the information, events, and network flow in system 100, generates deception information based on that analysis, and inserts the deception information into the system 100. In another example, deception system can operate in connection with Symantec Decoy Server, a honeypot intrusion detection system that detects the unauthorized access of information on system 100. In yet another example, deception system 114 can be multiple servers or workstations that simulate the information, events, and traffic between collaborating systems 102, 104, and 106.

[0037] In some embodiments, deception system 114 can also include one or more decoy servers and workstations that are created on-demand on actual servers and workstations (e.g., collaborating systems 102, 104, and 106) to create a realistic target environment. For example, deception infrastructure 114 can include dedicated virtual machines that can run on actual end-user workstations (e.g., one of collaborating systems 102, 104, and 106) by using hardware virtualization techniques.

[0038] In some embodiments, deception system 114 can record information, events, and network flow in system 100. For example, deception system 114 can monitor the execution of scripts containing sequences of traffic and events to observe natural performance deviations of communications network 108 and collaborating systems 102, 104, and 106 from the scripts, as well as the ability to distinguish such natural performance deviations from artificially induced deviations. In response, deception system 114 can generate believable decoy information.

[0039] It should be noted that decoy information can include any suitable data that is used to trap attackers (e.g., human agents or their system, software proxies, etc.) and/or the malware. Decoy information can include user behavior at the level of network flows, application use, keystroke dynamics, network flows (e.g.,

collaborating system 102 often communicates with collaborating system 104), registry-based activity, shared memory activity, etc. For example, decoy information can be a copy of an actual document on the system but with changed dates and times. In another example, decoy information can be a copy of a password file on the system with changed passcodes. Decoy information that is generated based on actual information, events, and flows can steer malware that is seeking to access and/or misuse the decoy information to deception system 114. Decoy information can assist in the identification of malicious/compromised computers (e.g., malicious/compromised computer 110), internal intruders (e.g., rogue users), or external intruders (e.g., external system 116).

[0040] In some embodiments, however, deception system 114 can be designed to defer making public the identity of a potential attacker or a user suspected of conducting unauthorized activities until sufficient evidence connecting the user with the suspected activities is collected. Such privacy preservation can be used to ensure that users are not falsely accused of conducting unauthorized activities. For example, if a user mistakenly opens a document containing decoy information, the user can be flagged as a potential attacker. In addition, the deception system or any other suitable monitoring application can monitor the potential attacker to determine whether the potential attacker performs any other unauthorized activities. Alternatively, a profile can be created that models the intent of the potential attacker. The profile can include information on, for example, registry-based activities, shared memory (DLL) activities, user commands, etc.

[0041] It should be noted that, in some embodiments, decoy information can be difficult to distinguish from actual information used in the system. For example, decoy information can be generated to appear realistic and indistinguishable from actual information used in the system. To entice a sophisticated and knowledgeable attacker, the decoy information is emulated or modeled such that a threat or an attacker (e.g., rootkits, malicious bots, keyloggers, spyware, malware, inside attacker, etc.) cannot discern the decoy information from actual information, events, and traffic on system 100.

[0042] As shown in FIG. 2, an original document 202 and a decoy document with an embedded beacon 204 are provided. Although document 204 is embedded with a hidden beacon (e.g., embedded code, watermark code, executable code, etc.),

there are no discernible changes between the original document 202 and the decoy document 204. In some embodiments, some of the content within decoy document 204 can be altered. For example, private information, such as name, address, and social security number, can be altered such that decoy document 204 is harmless if accessed and/or retrieved by an attacker.

[0043] In some embodiments, deception system 114 can include a surrogate user bot that appears to the operating system, applications, and embedded malicious code as an actual user on system 100. Using a surrogate user bot along with a virtualization layer beneath each operating system and a monitoring environment, the surrogate user bot can follow scripts to send events through virtualized keyboard and mouse drivers, open applications, search for messages, input responses, navigate an intranet, cut and paste information, etc. The surrogate user bot can display the results of these events to virtualized screens, virtualized printers, or any other suitable virtualized output device. In some embodiments, the surrogate user bot can be used to post decoy information to blog-style web pages on a decoy service such that the blog, while visible to malware, potential intruders, and potential attackers, is not visible to users of system 100 that do not look for the decoy information using inappropriate approaches.

[0044] In some embodiments, deception system 114 can discover the identity and/or the location of attacking computer systems (e.g., attacking computer system 116). Deception system 114 can also discover the identity and/or the location of attackers or external attacking systems that are in communication with and/or in control of the malware. For example, a single computer can contain embedded decoy information, such as a document with a decoy username and password. A server, such as a web server, that identifies failed login attempts using the decoy username and password can receive the IP address and/or other identifying information relating to the attacking computer system along with the decoy username and password. Alternatively, the server can inform the single computer that the document containing the decoy username and password has been exfiltrated.

[0045] In some embodiments, decoy information can be used to confuse and/or slow down attacking computer system 116. For example, attacking computer system 116 can be forced to spend time and energy obtaining information and then sorting through the collected information to determine actual information from decoy

information. In another example, the decoy information can be modeled to contradict the actual or authentic data on system 100, thereby confusing attacking computer system 116 or the user of attacking computer system 116 and luring the user of attacking computer system 116 to risk further actions to clear the confusion.

[0046] In accordance with some embodiments, trap-based defenses can be provided using a process 300 as illustrated in FIG. 3. As shown, information, events, and network flows in the operating environment can be monitored at 302. For example, deception system 114 of FIG. 1 monitors user behavior at the level of network flows, application use, keystroke dynamics, network flows (e.g., collaborating system 102 often communicates with collaborating system 104), registry-based activity, shared memory activity, etc. FIG. 4 shows examples of actual Simple Mail Transfer Protocol (SMTP) traffic 402 and Post Office Protocol (POP) traffic 404 that can be monitored. In some embodiments, deception system 114 uses a monitoring application (e.g., a network protocol analyzer application, such as Wireshark) to monitor and/or analyze network traffic.

[0047] Referring back to FIG. 3, at 304, decoy information that is based at least in part on the monitored information, events, and network flows is generated. As described previously, decoy information can include any suitable data that is used to trap attackers and/or the malware. Decoy information can include user behavior at the level of network flows, application use, keystroke dynamics, network flows (e.g., collaborating system 102 often communicates with collaborating system 104), a sequence of activities performed by users on a collaborating system, a characterization of how the user performed the activities on the collaborating system, etc. For example, decoy information can be a copy of an actual document on the system but with changed dates and times. In another example, decoy information can be a copy of a password file on the system with changed passwords.

[0048] Illustrative examples of decoy information are shown in FIGS. 2 and 5. As shown in FIG. 2, decoy information can be a decoy tax document. As shown in FIG. 5, decoy SMTP traffic 502 and decoy POP traffic 404 based upon the actual SMTP traffic 402 and actual POP traffic 404 of FIG. 4, respectively, are generated. The decoy traffic shows decoy account usernames, decoy account passwords, decoy media access control (MAC) addresses, modified IP addresses, modified protocol

commands, etc. The decoy information can be used to entice attackers and/or malware seeking to access and/or misuse the decoy information.

[0049] It should be noted that, in order to generate decoy information at 304 that entices a sophisticated and knowledgeable attacker, the decoy information is emulated or modeled such that a threat or an attacker (e.g., rootkits, malicious bots, keyloggers, spyware, malware, etc.) cannot discern the decoy information from actual information, events, and traffic on the system. For example, the decoy information is modeled such that the decoy information is believable and indistinguishable from actual information.

[0050] In some embodiments, actual information, events, and network flows in the operating environment are recorded. For example, domain name server (DNS) name, Internet Protocol (IP) addresses of collaborating systems 102, 104, and 106 (FIG. 1), authentication credentials (e.g., a password), and the data content of the traffic (e.g., documents and email messages) are recorded. In another example, keyboard events related to an application (e.g., web browser) that indicates the input of a username and a password combination or a URL to a web server are recorded. In yet another example, network traffic containing particular protocols of interest (e.g., SMTP, POP, File Transfer Protocol (FTP), Internet Message Access Protocol (IMAP), Hypertext Transfer Protocol (HTTP), etc.) can be recorded. In response to recording the actual information, the decoy information can be modeled and generated based on the recorded information.

[0051] It should be noted that particular environments with privacy concerns can record a specific sample of information, events, and traffic and generate decoy information based on the recorded sample.

[0052] In some embodiments, the decoy information can be a modified version of the actual information, where the actual information is replicated and then the original content of the actual information is modified. For example, the date, time, names of specific persons, geographic places, IP addresses, passwords, and/or other suitable content can be modified (e.g., changed, deleted, etc.) from the actual information. In another example, the source and destination MAC addresses, the source and destination IP addresses, and particular tagged credentials and protocol commands can be modified from the actual information. Such modified content

renders the content in the decoy information harmless when the decoy information is accessed and/or executed by a potential attacker.

[0053] In some embodiments, in addition to modifying the content of the actual information, additional content can be inserted into the decoy information to attract attackers and/or malware. For example, keywords or attractive words, such as "confidential," "top secret," and "privileged," can be inserted into the decoy information to attract attackers and/or malware (e.g., a network sniffer) that are searching for particular keywords.

[0054] In some embodiments, publicly available documents that can be obtained using search engines, such as www.google.com and www.yahoo.com, can be used to generate decoy information. For example, the decoy information can be generated based on actual information acquired from human generated sources, such as web blogs.

[0055] In some embodiments, existing historical information, such as previously recorded network data flows, can be used to create traceable, synthetic decoy information. Using existing historical information can mitigate the risk of detection by attackers and/or malware, such as network sniffers, because the flow of the decoy information generated using the historical information can be similar to prior traffic that the network sniffers have seen.

[0056] It should be noted that use of the historical information is localized to specific collaborating system or specific network segments to inhibit the exposure of sensitive information. For example, recorded historical information in one subnet is not used in another subnet to avoid exposing sensitive information that would otherwise remain hidden from malware located in one of the subnets.

[0057] In some embodiments, snapshots of the collaborating system's environment can be taken at given times (e.g., every month) to replicate the environment, including any hidden malware therein. The snapshots can be used to generate decoy information for the collaborating system.

[0058] In some embodiments, decoy information can be generated in response to a request by the user. For example, a system administrator or a government intelligence officer can fabricate decoy information (e.g., decoy documents) that is attractive to malware or potential attackers. Malware that is designed to spy on the network of a government intelligence agency can be attracted to different types of

information in comparison to malware that is designed to spy on the corporate network of a business competitor. Accordingly, a user (e.g., government intelligence officer) can create tailored decoy information, such as a top secret jet fighter design document or a document that includes a list of intelligence agents.

[0059] As shown in FIGS. 6 and 7, a website or any other suitable interface can be provided to a user for generating and/or downloading decoy information. In FIG. 6, the website requests that the user register with a legitimate email address. In response to registering with the website and logging into the website, the website allows the user to generate and/or download decoy information as shown in FIG. 7. As shown, display 700 provides the user with fields 702, 704, 706, and 708 for generating decoy documents. Fields 702 allow the user to select a particular type of decoy document to generate (e.g., a Word document, a PDF document, an image document, a URL link, etc.). Fields 704 allow the user to select a theme for the decoy document (e.g., a shopping list, a lost credit card document, a budget report, a personal document, a tax return document, an eBay receipt, a bank statement, etc.). Fields 706 and 708 allow the user to input a particular user and/or company involved in the decoy document. Alternatively, fields 706 and 708 allow the user to indicate that the website can select a random user and/or company for inclusion in the decoy document. In response to the user selecting, for example, a generate button 710 (or any other suitable user interface), the website generates a decoy document and provides the decoy document to the user. For example, the user can download the decoy document from the website.

[0060] Illustrative examples of generated decoy media and decoy documents are shown in FIGS. 2 and 8. As shown in FIG. 2, decoy information can be a decoy tax document. As shown in FIG. 8, the website has generated three decoy documents – e.g., a decoy lost credit card letter 802 and a decoy Microsoft Excel file 804 that includes decoy customer information (e.g., names, addresses, credit card numbers, tracking numbers, credit card expiration dates, etc.).

[0061] It should be noted that, in some embodiments, the website can instruct the user to place the decoy document in a particular folder. For example, as shown in FIG. 7, the website recommends that the user place the document in a location, such as the "My Documents" folder or any other suitable folder (e.g., a "Tax" folder, a "Personal" folder, a "Private" folder, etc.).

[0062] It should also be noted that, in some embodiments, the website can refresh decoy information such that attackers cannot identify particular static documents and information as decoy information.

[0063] In some embodiments, the website can provide a user with information that assists the user to more effectively deploy the decoy media. The website can prompt the user to input information suggestive of where the deception system or any other suitable application can place the decoy media to better attract potential attackers. For example, the user can indicate that the decoy information or decoy media be placed in the "My Documents" folder on collaborating system. In another example, the website can instruct the user to create a folder for the insertion of decoy media, such as a "My Finances" folder or a "Top Secret" folder.

[0064] In some embodiments, the website can request to analyze the system for placement of decoy information. In response to the user allowing the website to analyze the user's computer, the website can provide the user with a list of locations on the user's computer to place decoy information (e.g., the "My Documents" folder, the "Tax Returns" folder, the "Temp" folder associated with the web browser, a password file, etc.). In some embodiments, in response to the user allowing the website to analyze the user's computer, the website can record particular media from the user's computer and generate customized decoy media. In some embodiments, in response to the user allowing the website to analyze the user's computer, the website can provide a list of recommended folders to place decoy media.

[0065] In some embodiments, the website can transmit notifications to the user in response to discovering that the decoy media has been accessed, transmitted, opened, executed, and/or misused. For example, in response to an attacker locating and opening a decoy document that includes decoy credit card numbers, the website can monitor for attempts by users to input a decoy credit card number. In response to receiving a decoy credit card number, the website can transmit an email, text message, or any other suitable notification to the user. In another example, the decoy information can include decoy usernames and/or decoy passwords. The website can monitor for failed login attempts and transmit an email, text message, or any other suitable notification to the user when an attacker uses a decoy username located on the user's computer.

[0066] Referring back to FIG. 3, in some embodiments, one or more beacons can be associated with and/or embedded into the generated decoy information at 306. Next, at 308, the decoy information along with the embedded beacons are inserted into the operating environment.

[0067] A beacon can be any suitable code (executable or non-executable) that can be inserted or embedded into decoy information and that assists in indicating that decoy information has been accessed, transmitted, opened, executed, and/or misused. In some embodiments, the beacon is executable code that can be configured to transmit signals (e.g., a ping) to indicate that the decoy information has been accessed, transmitted opened, executed, and/or misused. For example, in response to an attacker opening a decoy document, the embedded beacon (e.g., in the form of a macro) transmits information about the attacker to a website. In some embodiments, the beacon is embedded code or watermark code (e.g., an embedded decoy or decoy financial information) that is detected upon attempted use. For example, the beacon can be a decoy token (e.g., a decoy login) that is detected when the attacker attempts to misuse the decoy token (e.g., by entering the decoy login). In some embodiments, the beacon is an embedded mark or code hidden in the decoy media or document that is scanned for on any (egress) network connections from a host. Alternatively, the beacon is an embedded mark or code hidden in the decoy media or document that is scanned for in memory whenever a file is loaded into an application, such as an encryption application.

[0068] It should be noted that, in some embodiments, the beacon can use routines (e.g., a Common Gateway Interface (CGI) script) to instruct another application on the attacker computer system to transmit a signal to indicate that the decoy information has been accessed, transmitted, opened, executed, and/or misused. For example, when the decoy document is opened by an attacker, the embedded beacon causes the attacker computer system to launch a CGI script that notifies a beacon website. In another example, when a decoy Microsoft Word document is opened by an attacker, the embedded beacon uses a CGI route to request that Microsoft Explorer transmit a signal over the Internet to indicate that the decoy document has been exfiltrated.

[0069] An illustrative example of the execution of an embedded beacon in a decoy document is shown in FIG. 9. As shown, in response to the attacker opening

decoy tax document 204 (FIG. 2), the Adobe Acrobat software application runs a Javascript function that displays window 902. Window 902 requests that the attacker allow a connection to a particular website. In response to selecting the "Allow" button or any other suitable user interface, a signal to the website (adobe-fonts.cs.columbia.edu) with information relating to the exfiltrated document and/or information relating to the attacker is transmitted (as shown in FIG. 10).

[0070] It should also be noted that multiple beacons can be placed in decoy media or any other suitable decoy information to detect access or attempted exfiltration by an inside attacker, an external attacker, or malware. For example, a decoy document can include an embedded beacon that automatically transmits a signal to a beacon website in response to opening the decoy document and multiple embedded decoy tokens that are detected in response to using any one of the decoy tokens. If the decoy document is accessed by the attacker when Internet access is not available, the beacons in the form of decoy tokens are detected in response to the attacker attempting to use the decoy tokens from the decoy document.

[0071] In some embodiments, beacon signals can include information sufficient to identify and/or trace the inside attacker, external attacker, or malware. Beacon signals can include the location of the attacker, the trail of the attacker, the unauthorized actions that the attacker has taken, etc. For example, in response to opening a decoy document, the embedded beacon can automatically execute and transmit a signal to a monitoring website. FIG. 11 provides an example of a website that collects signals from one or more beacons. As shown, the signal (e.g., the beacon ping) can include information relating to the attacker, such as the IP address, the exfiltrated document, and the time that the attacker opened the document. In another example, decoy login identifiers to particular servers can be generated and embedded in decoy documents. In response to monitoring a daily feed list of failed login attempts, the server can identify exfiltrated documents.

[0072] In some embodiments, the beacon can track and/or identify the attacker by using a vendor supplied serial number extracted from a vendor application on the attacker computer system (e.g. stored in the Microsoft Windows Registry). For example, the serial number of a media player licensed to the attacker can be used to identify the attacker from business records at the time the attacker purchased and/or registered the media player. In another example, applications loaded on the attacker

computer system can be used as an index into a vendor database to identify who or which entity purchased the attacker computer system.

[0073] In some embodiments, a beacon or a decoy token (e.g., a decoy username and password) that is inserted or injected into decoy media (e.g., decoy FTP login sessions) can be updated, changed, and/or varied over time. For example, at a given time (e.g., every ten seconds), the decoy token within the decoy media can be changed. By tracking the different decoy tokens over time and detecting that the decoy token has been used (e.g., an attempted login using the decoy username and password), the system can determine when the exfiltration or leakage occurred. For example, the system can access a lookup table of decoy tokens and, based on the decoy token that was used to login to the system, the system can determine the time at which the exfiltration occurred.

[0074] In some embodiments, the beacon can be a portion of code embedded in documents or other media in a manner that is not obvious to malware or an attacker. The beacon can be embedded such that an attacker is not aware that the attacker has been detected. For example, referring back to FIG. 9, the Javascript function is used to hide the embedded beacon, where the displayed Javascript window requests that the attacker execute the beacon code. In another example, the beacon can be embedded as a believable decoy token.

[0075] It should be noted that, in some embodiments, the creator or the producer of the application that opens the decoy information may provide the capability within the application to execute embedded beacons. For example, an application creator that develops a word processing application may configure the word processing application to automatically execute embedded beacons in decoy information opened by the word processing application. Accordingly, the application automatically executes the beacon code and does not request that the attacker execute the beacon code.

[0076] It should also be noted that document formats generally consist of a structured set of objects of any type. The beacon can be implemented using obfuscation techniques that share the appearance of the code implementing the beacon to appear with the same statistical distribution as the object within which it is embedded. Obtaining the statistical distribution of files is described in greater detail in, for example, Stolfo et al., U.S. Patent Publication No.2005/0265311 A1, published

December 1, 2005, Stolfo et al., U.S. Patent Publication No. 2005/0281291 A1, published December 22, 2005, and Stolfo et al., U.S. Patent Publication No. 2006/0015630 A1, published January 19, 2006, which are hereby incorporated by reference herein in their entireties.

[0077] In some embodiments, each collaborative system (e.g., collaborative systems 102, 104, and 106) can designate a particular amount of storage capacity available for decoy information. For example, a collaborative system can indicate that 50 megabytes of storage space is available for decoy information. In some embodiments, decoy information can be distributed even among the collaborative systems in the network. For example, in response to generating 30 megabytes of decoy information, each of the three collaborative systems in the network receives 10 megabytes of decoy information. Alternatively, collaborative systems can receive any suitable amount of decoy information such that the decoy information appears believable and cannot be distinguished from actual information. For example, deception system 114 of FIG. 1 can generate decoy information based on the actual information (e.g., documents, files, e-mails, etc.) on each collaborative system. In another example, deception system 114 can generate a particular amount of decoy information for each collaborative system based on the amount of actual information is stored on each collaborative system (e.g., 10% of the actual information).

[0078] In accordance with some embodiments, decoy information with embedded beacons are implemented using a process 1200 as illustrated in FIG. 12. Decoy information can assist in the identification of malicious/compromised computers (e.g., malicious/compromised computer 110 of FIG. 1), internal intruders (e.g., rogue users), or external intruders.

[0079] As shown, at 1202, once decoy information is inserted into the operating environment, a signal from an embedded beacon in a particular piece of decoy information can be received in response to detecting activity of the particular piece of decoy information. The embedded beacon can be configured to transmit signals to indicate that the particular piece of decoy information has been accessed, opened, executed, and/or misused. For example, in response to opening, downloading, and/or accessing the document or any other suitable media that includes the decoy information, the embedded beacon can be automatically executed to transmit a signal that the decoy information has been accessed.

[0080] In some embodiments, beacons can be implemented in connection with a host-based monitoring application (e.g., an antivirus software application) that monitors the beacons or beacon signatures. For example, the host-based monitoring application can be configured to transmit signals or an alert when it detects specific signatures in documents. By embedding specific beacon signatures in the decoy documents, the software application can detect and receive beacon signals each time the decoy documents are accessed, opened, etc. Information about the purloined document can be uploaded by the monitoring application.

[0081] At 1204, in some embodiments, the beacon signal can include information sufficient to identify the location of the attacker and/or monitor the attacker. Beacon signals can include the location of the attacker, the trail of the attacker, the unauthorized actions that the attacker has taken, etc. In some embodiments, beacon signals can include information identifying the attacker computer system (e.g., an IP address) that received and/or accessed the decoy information through an exfiltration channel.

[0082] In some embodiments, the beacon embedded in the decoy information can indicate the presence of an attacker to a user (e.g., a user of collaborative system 102, 104, or 106). For example, the decoy information can be a decoy login and a decoy password that is capable of detecting an attacker and monitoring the unauthorized activities of the attacker. In response to the decoy login and/or the decoy password being used on a website, the web server can send a notification to the user that the system of the user has been compromised.

[0083] In some embodiments, the beacon embedded in the decoy information can record an irrefutable trace of the attacker when the decoy information is accessed or used by the attacker. For example, the deception system 114 of FIG. 1 uses a back channel that an attacker cannot disable or control. A back channel can notify a website or any other suitable entity that the decoy information (e.g., decoy passwords) is being used. Using the back channel, the website of a financial institution can detect failed login attempts made using passwords that were provided by a decoy document or a decoy network flow. Accordingly, it would be difficult for an attacker to deny that the attacker obtained and used the decoy information. Alternatively, in response to opening the decoy information in the decoy media (e.g., a decoy document), the embedded beacon can transmit a signal to the website of the financial institution.

[0084] For example, in some embodiments, the beacon embedded in the decoy information can transmit a signal to a website that logs the unauthorized access of the decoy information by an attacker. The user of a collaborative system can access the website to review the unauthorized access of the decoy information to determine whether the access of the decoy information is an indication of malicious or nefarious activity. In some embodiments, the website can log information relating to the attacker for each access of the decoy information.

[0085] At 1206, in some embodiments, with the use of other applications, the malware can be removed in response to receiving the information from the embedded beacon. For example, in response to identifying that malicious code in a particular document is accessing the decoy information (or that an attacker is using the malicious code embedded in a particular document to access the decoy information), the beacon can identify the source of the malicious code and send a signal to a monitoring application (e.g., an antivirus application or a scanning application) that parses through the document likely containing the malicious code. In another example, the beacon can identify that malicious code lies dormant in the file store of the environment awaiting a future attack.

[0086] In accordance with some embodiments, decoy information with embedded beacons can transmit additional notifications and/or recommendations using a process 1300 as illustrated in FIG. 13.

[0087] As shown, at 1302, once decoy information is inserted into the operating environment, a signal from an embedded beacon in a particular piece of decoy information can be received in response to detecting activity of the particular piece of decoy information. The embedded beacon can be configured to transmit signals to indicate that the particular piece of decoy information has been accessed, opened, executed, and/or misused. For example, in response to opening, downloading, and/or accessing the document or any other suitable media that includes the decoy information, the embedded beacon can be automatically executed to transmit a signal that the decoy information has been accessed.

[0088] In some embodiments, in response to receiving a signal from a beacon, the actual information (e.g., the original document) associated with the decoy information can be determined at 1304. For example, in response to receiving a signal from a beacon, the deception system can determine the actual information that

the decoy information was based on and determine the computing system where the actual information is located. In response, at 1306, the collaborative system that has the actual information can be alerted or notified of the accessed decoy information. In some embodiments, the collaborative system can be notified of the decoy information that was accessed, information relating to the computer that accessed, opened, executed, and/or misused the decoy information (or the media containing the decoy information), etc. For example, the deception system can transmit the user name and the IP address of the attacker computer system. In another example, the deception system can transmit, to the computing system, a recommendation to protect the actual information or the original document that contains the actual information (e.g., add or change the password protection).

[0089] It should be noted that, in some embodiments, deception system 114 or any other suitable system can be designed to defer making public the identity of a potential attacker or a user suspected of conducting unauthorized activities until sufficient evidence connecting the user with the suspected activities is collected. Such privacy preservation can be used to ensure that users are not falsely accused of conducting unauthorized activities.

[0090] In some embodiments, the efficacy of the generated decoy media (or decoy information) can be measured by monitoring usage of the decoy information. For example, for a website of a financial institution, the efficacy of the generated decoy media can be measured by monitoring the number of failed login attempts (e.g., on a website, daily feed, secure shell login accounts, etc.). In some embodiments, the efficacy of the generated decoy media can be measured by monitoring egress traffic or file system access. In some embodiments, the efficacy of the generated decoy media can be used to generate reports on the security of a collaborative system or any other suitable device.

[0091] In accordance with some embodiments, decoy information can be inserted into a particular software application. For example, decoy information can be inserted specifically into the Microsoft Outlook application. The decoy information can be inserted as decoy emails, decoy notes, decoy email addresses, decoy address book entries, decoy appointments, etc. In some embodiments, decoy email messages can be exchanged between decoy accounts to expose seemingly confidential information to malware or an attacker searching for particular keywords. Any attempt

by the malware or an attacker using an external system in communication with the malware to access the decoy information can then be quickly detected. Evidence indicative of unauthorized activities can be collected and studied. For example, a deviation from the pre-scripted decoy traffic, unscripted access to decoy information, and/or various other suitable anomalous events can be collected.

[0092] In some embodiments, decoy information can be broadcast as decoy traffic. For example, collaborating systems or other suitable devices that have a wireless connection (e.g., a Wi-Fi-enabled computer) can broadcast decoy traffic. A notification can be sent to the collaborating system in response to another entity accessing the decoy traffic.

[0093] In some embodiments, decoy information can be inserted onto multiple devices. For example, a website can be provided to a user that places decoy information contained in decoy media on registered devices (e.g., the user's computer, the user's personal digital assistant, the user's set-top box, the user's cellular telephone, etc.). Once the decoy media is accessed, a notification can be sent to the user. It should be noted that, as decoy media generally does not have production value other than to attract malware and or potential attackers, activity involving decoy media is highly suggestive of a network compromise or other nefarious activity.

[0094] Accordingly, methods and systems are provided for providing trap-based defenses.

[0095] Although the invention has been described and illustrated in the foregoing illustrative embodiments, it is understood that the present disclosure has been made only by way of example, and that numerous changes in the details of implementation of the invention can be made without departing from the spirit and scope of the invention, which is only limited by the claims which follow. Features of the disclosed embodiments can be combined and rearranged in various ways.

What is claimed is:

1. A method for providing trap-based defenses, the method comprising:
generating decoy information;
embedding a beacon into the decoy information; and
inserting the decoy information with the embedded beacon into a computing environment, wherein the embedded beacon provides an indication that the decoy information has been accessed by an attacker.
2. The method of claim 1, wherein the beacon is a decoy token embedded in the decoy information.
3. The method of claim 2, further comprising monitoring for unauthorized usage of the decoy token.
4. The method of claim 1, wherein the beacon is executable code embedded in the decoy information.
5. The method of claim 4, further comprising executing the executable code in response to the attacker accessing the decoy information, wherein the executable code transmits a signal indicating that the decoy information has been accessed.
6. The method of claim 4, further comprising requesting that the attacker permit the execution of the executable code in response to the attacker accessing the decoy information.
7. The method of claim 1, further comprising:
receiving a signal from the embedded beacon in response to the attacker accessing the decoy information; and
transmitting a notification that provides information relating to the attacker.

8. The method of claim 7, further comprising automatically instructing another application to transmit a notification that provides information relating to the attacker.

9. The method of claim 1, further comprising:
detecting that decoy information with the embedded beacon has used;
and
transmitting a notification that provides information relating to the attacker.

10. The method of claim 1, further comprising monitoring for at least one signal transmitted from the embedded beacon.

11. The method of claim 1, further comprising embedding a plurality of beacons within the decoy information.

12. The method of claim 11, further comprising receiving a plurality of indications from the plurality of beacons in response to an attacker accessing the decoy information.

13. The method of claim 12, wherein at least one of the plurality of indications is a signal that the decoy information has been accessed.

14. The method of claim 12, wherein at least one of the plurality of indications is an indication that the decoy information has been used.

15. The method of claim 1, wherein the generated decoy information is based at least in part on actual information in the computing environment.

16. The method of claim 15, further comprising:
replicating the actual information to generate the decoy information;
and

inserting one or more keywords into the decoy information.

17. The method of claim 15, further comprising:
replicating the actual information that includes private information and public information; and
altering the private information to generate the decoy information.

18. The method of claim 1, further comprising receiving, from a user, at least one location within the computing environment to insert the decoy information.

19. The method of claim 1, further comprising tracing an attacker in response to receiving the signal from the embedded beacon.

20. The method of claim 1, further comprising analyzing the computing environment to determine where to insert the decoy information.

21. The method of claim 1, wherein the embedded beacon is configured to operate in connection with a monitoring application, and wherein the monitoring application monitors the computing environment for the signal from the embedded beacon.

22. The method of claim 21, wherein the monitoring application is a host-based monitoring application.

23. The method of claim 21, wherein the monitoring application is an antivirus application.

24. The method of claim 1, wherein the embedded beacon is activated in response to opening the decoy media containing the decoy information.

25. The method of claim 24, wherein the embedded beacon is activated when the decoy media is one of: rendered, copied, transmitted, opened, and executed.

26. The method of claim 1, further comprising:
generating a plurality of beacons;
updating the embedded beacon with at least one of the plurality of beacons at a given time, wherein the at least one of the plurality of beacons is associated with the given time; and
determining a time that the attacker accessed the decoy document based on the at least one of the plurality of beacons and the associated time.
27. A system for providing trap-based defenses, the system comprising:
a processor that:
generates decoy information;
embeds a beacon into the decoy information; and
inserts the decoy information with the embedded beacon into a computing environment, wherein the embedded beacon provides an indication that the decoy information has been accessed by an attacker.
28. The system of claim 27, wherein the beacon is a decoy token embedded in the decoy information.
29. The system of claim 28, wherein the processor is further configured to monitor for unauthorized usage of the decoy token.
30. The system of claim 27, wherein the beacon is executable code embedded in the decoy information.
31. The system of claim 30, wherein the processor is further configured to execute the executable code in response to the attacker accessing the decoy information, wherein the executable code transmits a signal indicating that the decoy information has been accessed.
32. The system of claim 30, wherein the processor is further configured to request that the attacker permit the execution of the executable code in response to the attacker accessing the decoy information.

33. The system of claim 27, wherein the processor is further configured to:
receive a signal from the embedded beacon in response to the attacker
accessing the decoy information; and
transmit a notification that provides information relating to the
attacker.

34. The system of claim 33, wherein the processor is further configured to
automatically instruct another application to transmit a notification that provides
information relating to the attacker.

35. The system of claim 27, wherein the processor is further configured to:
detect that decoy information with the embedded beacon has used; and
transmit a notification that provides information relating to the
attacker.

36. The system of claim 27, wherein the processor is further configured to
monitor for at least one signal transmitted from the embedded beacon.

37. The system of claim 27, wherein the processor is further configured to
embed a plurality of beacons within the decoy information.

38. The system of claim 37, wherein the processor is further configured to
receive a plurality of indications from the plurality of beacons in response to an
attacker accessing the decoy information.

39. The system of claim 38, wherein at least one of the plurality of
indications is a signal that the decoy information has been accessed.

40. The system of claim 38, wherein at least one of the plurality of
indications is an indication that the decoy information has been used.

41. The system of claim 27, wherein the processor is further configured to generate the decoy information based at least in part on actual information in the computing environment.

42. The system of claim 41, wherein the processor is further configured to: replicate the actual information to generate the decoy information; and insert one or more keywords into the decoy information.

43. The system of claim 41, wherein the processor is further configured to: replicate the actual information that includes private information and public information; and
alter the private information to generate the decoy information.

44. The system of claim 27, wherein the processor is further configured to receive, from a user, at least one location within the computing environment to insert the decoy information.

45. The system of claim 27, wherein the processor is further configured to trace an attacker in response to receiving the signal from the embedded beacon.

46. The system of claim 27, wherein the processor is further configured to analyze the computing environment to determine where to insert the decoy information.

47. The system of claim 27, wherein the embedded beacon is configured to operate in connection with a monitoring application, and wherein the monitoring application monitors the computing environment for the signal from the embedded beacon.

48. The system of claim 47, wherein the monitoring application is a host-based monitoring application.

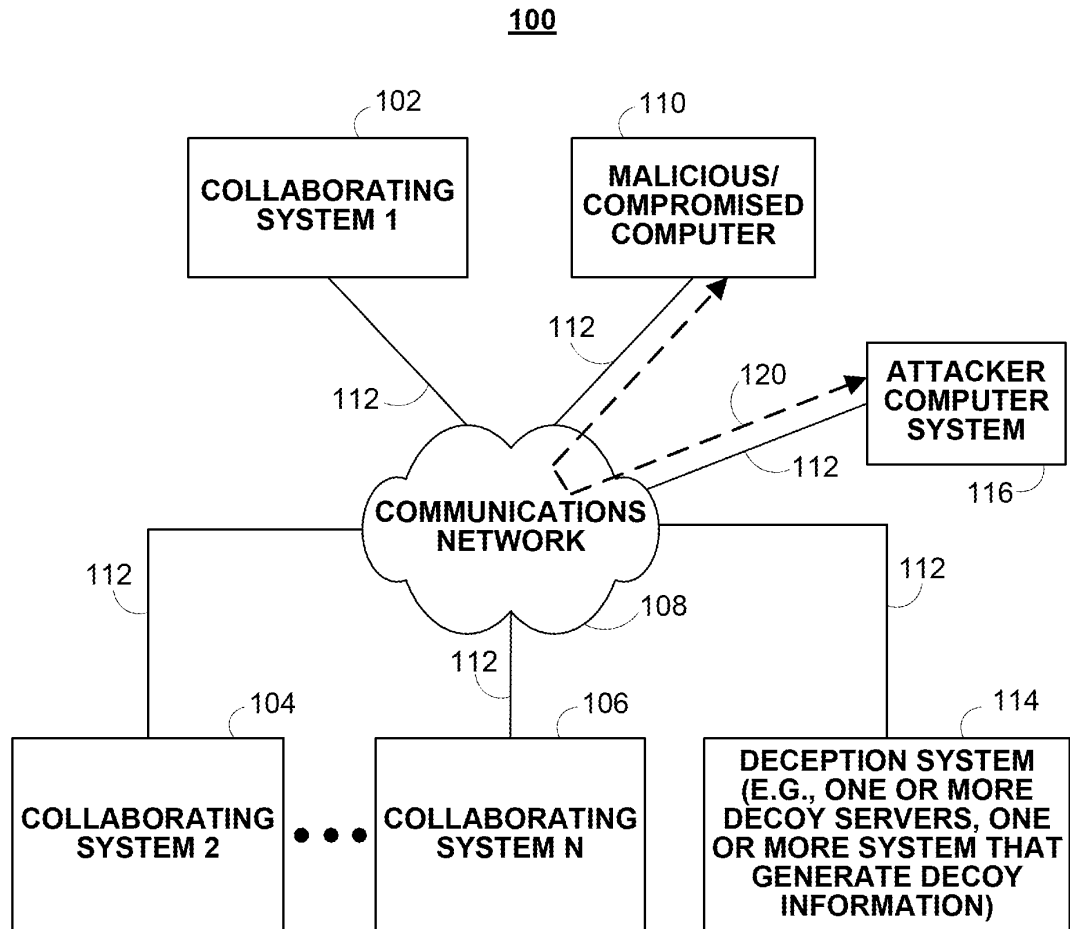
49. The system of claim 47, wherein the monitoring application is an antivirus application.

50. The system of claim 27, wherein the embedded beacon is activated in response to opening the decoy media containing the decoy information.

51. The system of claim 50, wherein the embedded beacon is activated when the decoy media is one of: rendered, copied, transmitted, opened, and executed.

52. The system of claim 27, wherein the processor is further configured to:
generate a plurality of beacons;
update the embedded beacon with at least one of the plurality of beacons at a given time, wherein the at least one of the plurality of beacons is associated with the given time; and
determine a time that the attacker accessed the decoy document based on the at least one of the plurality of beacons and the associated time.

1/13

**FIG. 1**

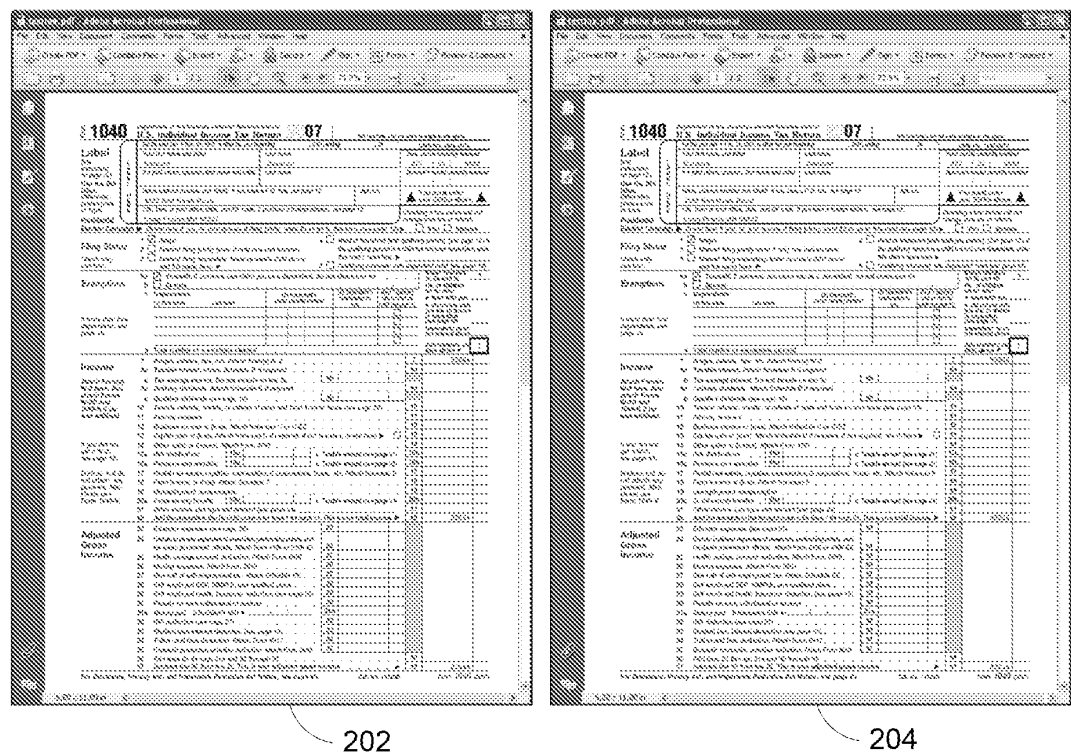
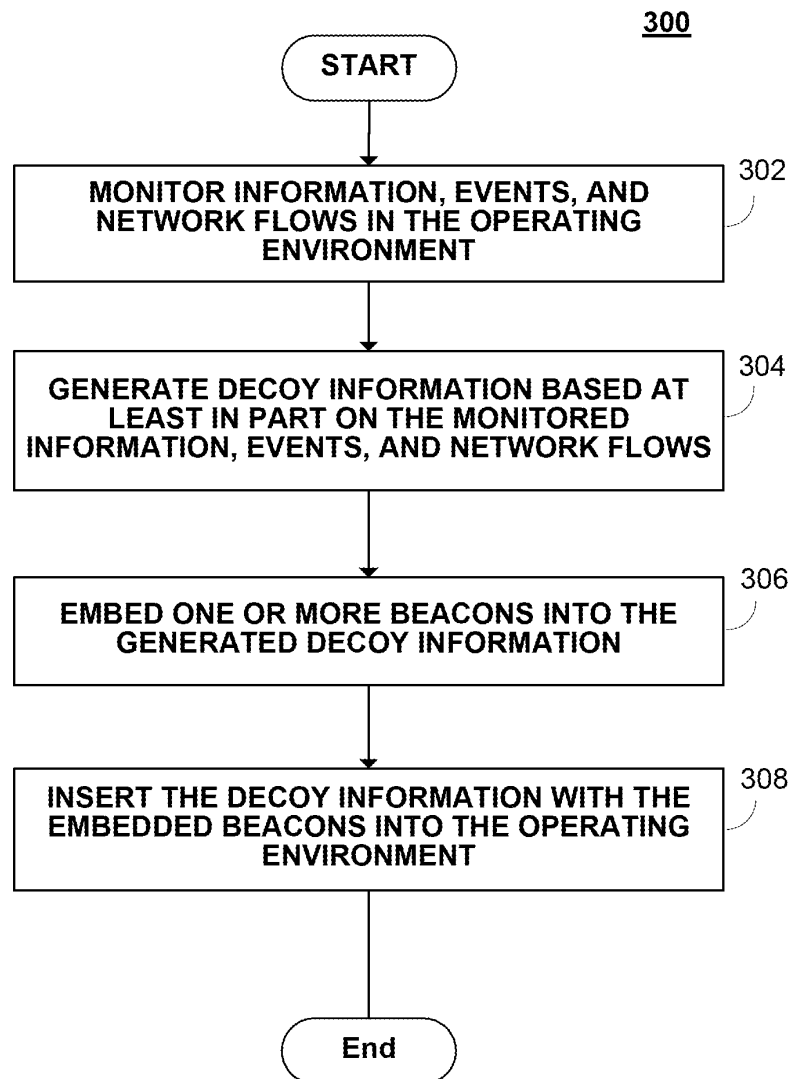


FIG. 2

3/13

**FIG. 3**

4/13

[illegible]

402

Time	Source	Destination	Add an expression to this filter string	Username
8.411091	198.39.247.72	63.248.76.16	TCP	42191 → 6031 [ACK] Seq=614261216 Len=0
8.412619	198.39.247.72	63.248.76.16	POP	Request: CAPX
8.425229	63.248.76.16	198.39.247.72	TCP	6031 → 42191 [ACK] Seq=614261216 Len=0
8.462611	63.248.76.16	198.39.247.72	POP	Response: +OK Capabilities list follows
8.462834	198.39.247.72	63.248.76.16	POP	Request: USER testfiltername1234
8.468789	63.248.76.16	198.39.247.72	TCP	6031 → 42191 [ACK] Seq=614261216 Len=0
8.469740	198.39.247.72	63.248.76.16	POP	Request: PASS myp@1234
8.512539	63.248.76.16	198.39.247.72	POP	Response: +OK
8.551705	198.39.247.72	63.248.76.16	TCP	42191 → 6031 [ACK] Seq=614261216 Len=0
8.629721	63.248.76.16	198.39.247.72	POP	Response: +OK ready
8.629774	198.39.247.72	63.248.76.16	TCP	42191 → 6031 [ACK] Seq=614261216 Len=0
8.629861	198.39.247.72	63.248.76.16	POP	Request: CAPX
8.734838	63.248.76.16	198.39.247.72	POP	Response: +OK
8.750525	198.39.247.72	63.248.76.16	POP	Request: LIST
8.814798	63.248.76.16	198.39.247.72	POP	Response: +OK 0 messages (0)
8.814955	198.39.247.72	63.248.76.16	POP	Request: UIDL
8.879825	63.248.76.16	198.39.247.72	POP	Response: +OK 0 messages (0)
8.880381	198.39.247.72	63.248.76.16	POP	Request: QUIT
8.887654	63.248.76.16	198.39.247.72	POP	Response: +OK Connection closed

1. From 30 (118 bytes on wire, 118 bytes captured)

Ethernet II, Src: Cisco-26:8c:00:18:00:26:26:80, Dst: Appleton-4c:08:a4:1609:17:72:4c:0d:04

Destination: Appleton-4c:08:a4:1609:17:72:4c:0d:04

Source: Cisco-26:8c:00:18:00:26:26:80

Type: IP (0x0080)

Internet Protocol, Src: 63.248.76.16 (63.248.76.16), Dst: 198.39.247.72 (198.39.247.72)

Version: 4

Protocol: 20, Len: 100

404

FIG. 4

5/13

Modified IPs for testing			Protocol	Info	Base 64 Encoded Tagged Credentials
Time	Source	Destination			
1.152472	0.0.0.0	1.1.1.1	TCP	38401 > satp [ACK] Seq=1	
1.198952	1.1.1.1	0.0.0.0	SMTP	Response: 220 col.columbia.edu	
1.198979	0.0.0.0	1.1.1.1	TCP	38401 > satp [ACK] Seq=1 Ack=23 Win=0 Len=0 TSv=2273821 TSER=17637	
1.199585	0.0.0.0	1.1.1.1	SMTP	Response: EHLO [0.0.0.0]	
1.246160	1.1.1.1	0.0.0.0	TCP	38401 > satp [ACK] Seq=23 Ack=18 Win=6608 Len=0 TSv=25317740	
1.253877	1.1.1.1	0.0.0.0			
1.254708	0.0.0.0	1.1.1.1			
1.300409	1.1.1.1	0.0.0.0			
1.302851	1.1.1.1	0.0.0.0			
1.303139	0.0.0.0	1.1.1.1			
1.379277	1.1.1.1	0.0.0.0	SMTP	Response: 250 ok	
1.379303	0.0.0.0	1.1.1.1	SMTP	Command: RCPT TO: anyone@test.com	
1.404726	1.1.1.1	0.0.0.0	SMTP	Response: 250 ok	
1.405458	0.0.0.0	1.1.1.1	SMTP	Response: DATA	
1.468267	1.1.1.1	0.0.0.0	SMTP	Response: 254 ok	
1.468489	0.0.0.0		SMTP	Message Body	
1.501826	1.1.1.1		TCP	satp > 38401 [ACK] Seq=0 Ack=18 Win=6608 Len=0 TSv=25317740	
1.552867	0.0.0.0	1.1.1.1	SMTP	Message Body	

Modified protocol commands to remove actual SMTP server

Base 64 Encoded Tagged Credentials

Bogus MACs

Frame 20 (74 bytes on wire, 74 bytes captured)

Ethernet II, Src: 08:02:05:0b:c4:77 (08:02:05:0b:c4:77), Dst: 08:f2:89:d4:f0:73 (08:f2:89:d4:f0:73)

Destination: 08:f2:89:d4:f0:73 (08:f2:89:d4:f0:73)

Source: 08:02:05:0b:c4:77 (08:02:05:0b:c4:77)

Type: IP (0x0800)

Internet Protocol, Src: 1.1.1.1 (1.1.1.1), Dst: 0.0.0.0 (0.0.0.0)

Version: 4

Header Length: 20, Options:

502

Arbitrary wireless CUIP as source			Protocol	Info	Decoy account username
Time	Source	Destination			
1.171231	128.59.23.85	128.59.23.85	PDP	Response: +OK (scrcp12) Mail	sa PDP3/PRIM server #135
1.171285	128.59.247.99	128.59.23.85	TCP	45191 > pop3 [ACK] Seq=1 Ack=0 Len=0 TSv=5592464 TSER=17637	
1.171396	128.59.247.99	128.59.23.85	PDP	Request: CAPA	
1.186299	128.59.23.85	128.59.247.99	TCP	pop3 > 45191 [ACK] Seq=52 Ack=7 Win=33384 Len=0 TSv=1763767179 TSER=5	
1.222596	128.59.23.85	128.59.247.99	PDP	Response: +OK Capability list follows	
1.222992	128.59.247.99	128.59.23.85	PDP	Request: USER decoyuser1234	
1.228728	128.59.23.85	128.59.247.99	TCP	pop3 > 45191 [ACK] Seq=250 Ack=27 Win=33384 Len=0 TSv=1763767183 TSER=	
1.228763	128.59.247.99	128.59.23.85	PDP	Request: PASS decoypassword	
1.272522	128.59.23.85	128.59.247.99	PDP	Response: +OK	
1.311738	128.59.247.99	128.59.23.85	TCP	45191 > pop3 [ACK] Seq=47 Ack=250 Win=6912 Len=0 TSv=5592605 TSER=176	
1.399059	128.59.23.85	128.59.247.99	PDP	Response: +OK read	
1.399715	128.59.247.99	128.59.23.85	TCP	45191 > pop3 [ACK] Seq=47 Ack=266 Win=6912 Len=0 TSv=5592683 TSER=176	
1.399858	128.59.247.99	128.59.23.85	PDP	Request: CAPA	
1.404249	128.59.23.85	128.59.247.99	PDP	Response: +OK Capability list follows	
1.404492	128.59.247.99		PDP	Request: UIDS	
1.574095	128.59.23.85		PDP	Response: +OK 0 messages (0)	
1.575925	128.59.247.99		PDP	Response: +OK 1	

Bogus MAC resembling a wireless linksys NIC

Decoy account password

Frame 1014 (74 bytes on wire, 74 bytes captured)

Ethernet II, Src: Linksys_08:f0:73 (08:06:25:04:f0:73), Dst: 08:02:05:0b:c4:77 (08:02:05:0b:c4:77)

Destination: 08:02:05:0b:c4:77 (08:02:05:0b:c4:77)

Source: Linksys_08:f0:73 (08:06:25:04:f0:73)

Type: IP (0x0800)

Internet Protocol, Src: 128.59.247.99 (128.59.247.99), Dst: 128.59.23.85 (128.59.23.85)

Transmission Control Protocol, Src Port: 45191 (45191), Dst Port: pop3 (110), Seq: 5, Len: 0

504

FIG. 5

6/13

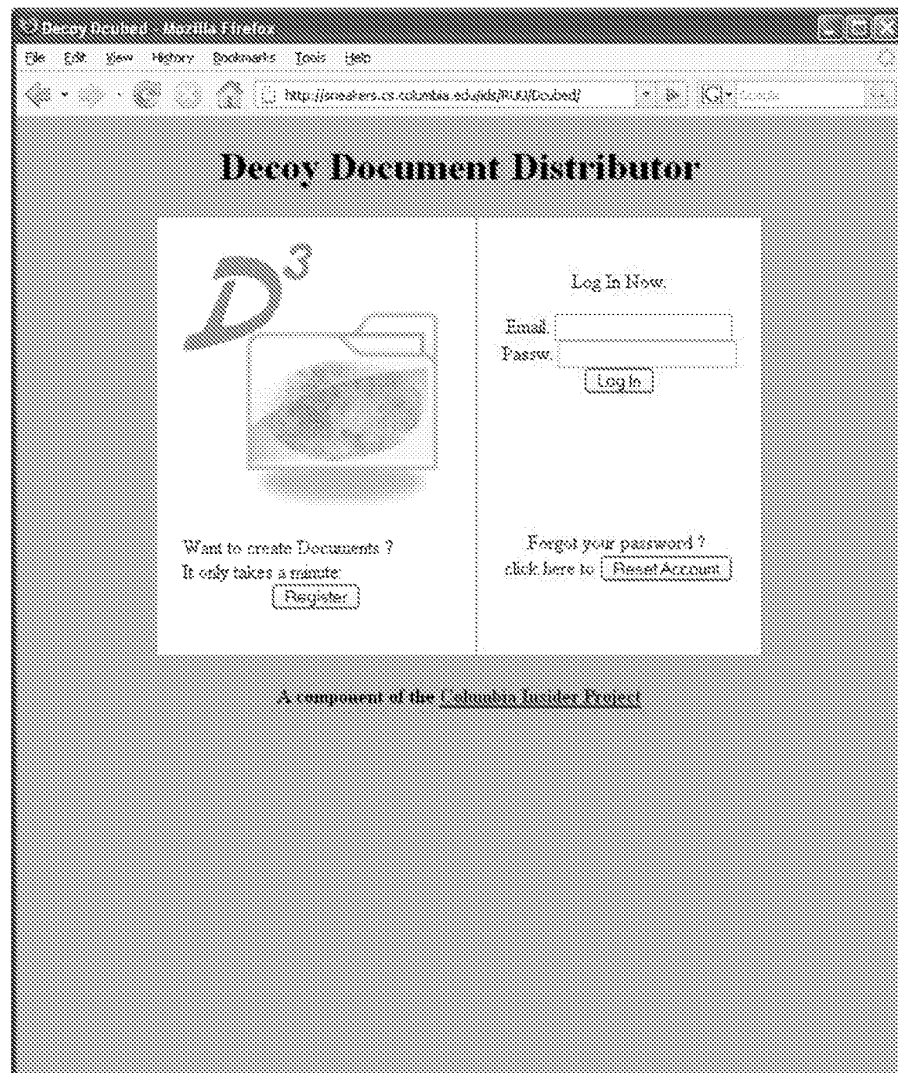


FIG. 6

7/13

700

Welcome to the Decoy Document Distributor

Create Decoy Document

Version 1.0 provides PDF documents.

You may opt to receive automatic warnings when your decoy documents have been stolen and accessed. After requesting and downloading a decoy document, we recommend you place the document in a conspicuous location such as in the "My Documents" folder or any other folder with an interesting name (Tax, Personal, Private, etc).

You may test your document by simply opening it.
You should receive an email warning message soon after.

Registered email: john.smith@email.com

Document Type	Theme
☐ Microsoft Word ☐ PDF ☐ Image ☐ URL LINK	☐ Shopping List ☐ Lost CC ☐ Budget ☐ Personal ☐ Tax Returns ☐ EBAY RCPT ☐ Bank Statement

☐ Random User ☐ Custom ☐ Random Company ☐ Custom
 User Involved: Corporation:

FIG. 7

9/13

204

The image shows a screenshot of a web browser window displaying the 2007 U.S. Individual Income Tax Return form (1040). The form is titled "U.S. Individual Income Tax Return" and includes fields for the taxpayer's name, social security number, and filing status. A security warning dialog box is overlaid on the form, displaying a warning icon and the text: "The document is trying to connect to http://adobe-fonts.cs.columbia.edu/cgi-bin/text.pl.cgi?document=pdf&pass=9c92ce63e9432b6f2c3c958d6437e042&title=Taxes&os=WIN&version=8.1%23FDF". The dialog box also includes a checkbox for "Remember my action for this site" and buttons for "Help", "Allow", and "Block".

Form 1040 U.S. Individual Income Tax Return

For the year Jan. 1-Dec. 31, 2007, or other tax year beginning 2007, ending 2007

Label (See instructions on page 12.) Use the IRS label. Otherwise, please print or type.

Presidential Election Campaign Check here if you, or your spouse, are eligible to claim the credit for federal tax on state and local taxes paid. ☐ **You** ☐ **Spouse**

Filing Status Check only one box.

1 ☐ **Single**

2 ☐ **Married filing jointly** (even if only one taxpayer has income)

3 ☐ **Married filing separately**. Enter spouse's SSN above and full name here.

Exemptions If more than four dependents, see page 10.

6a ☐ **Yourself** someone can claim you as a dependent, **do not** check box 6a.

b ☐ **Spouse**

c Dependents:

(1) First name	Last name	(2) Dependent's social security number	(3) Dependent's relationship to you (see page 10)	(4) If dependent is your child, enter date of birth (see page 10)

d Total number of exemptions claimed

7 Wages, salaries, tips, etc. Attach Form(s) W-2

Security Warning

The document is trying to connect to
<http://adobe-fonts.cs.columbia.edu/cgi-bin/text.pl.cgi?document=pdf&pass=9c92ce63e9432b6f2c3c958d6437e042&title=Taxes&os=WIN&version=8.1%23FDF>

If you trust the site, choose Allow. If you do not trust the site, choose Block.

☐ Remember my action for this site

Help Allow Block

FIG. 9

10/13

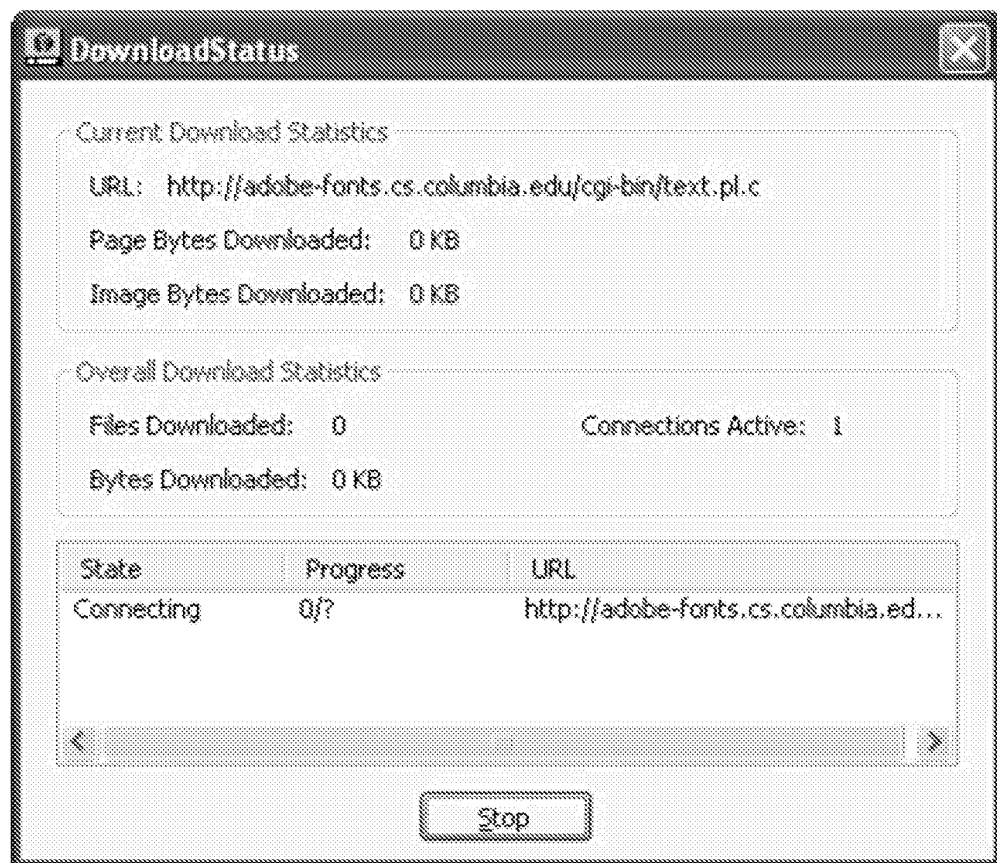


FIG. 10

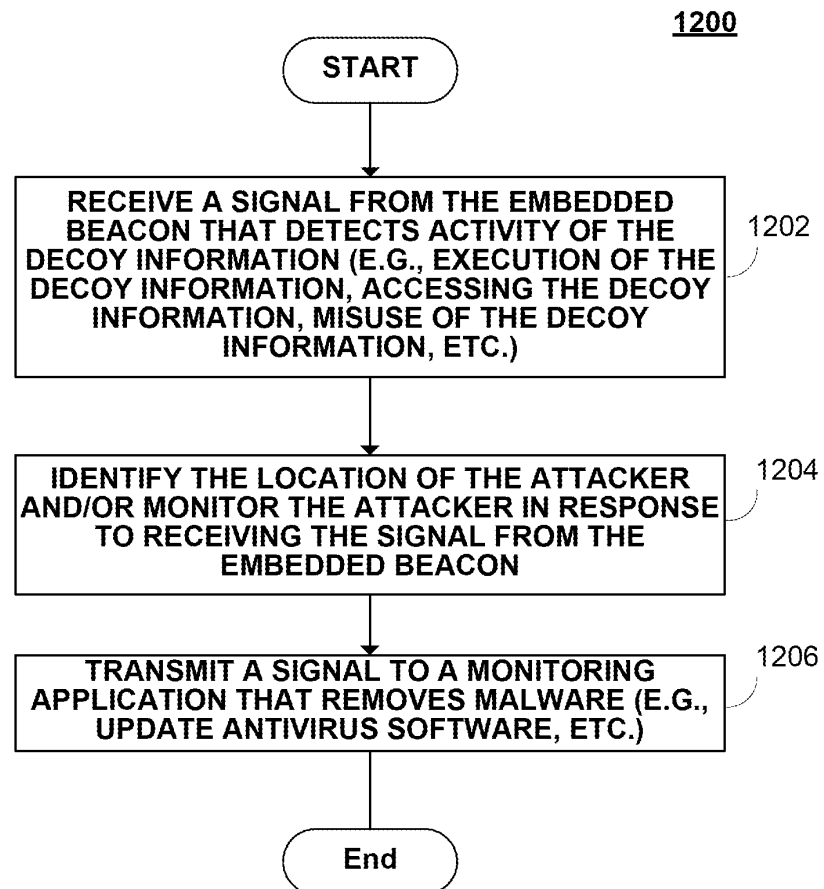
11/13

SONAR (v 0.4) Summary
Log file: logfile.txt
Number of records: 164

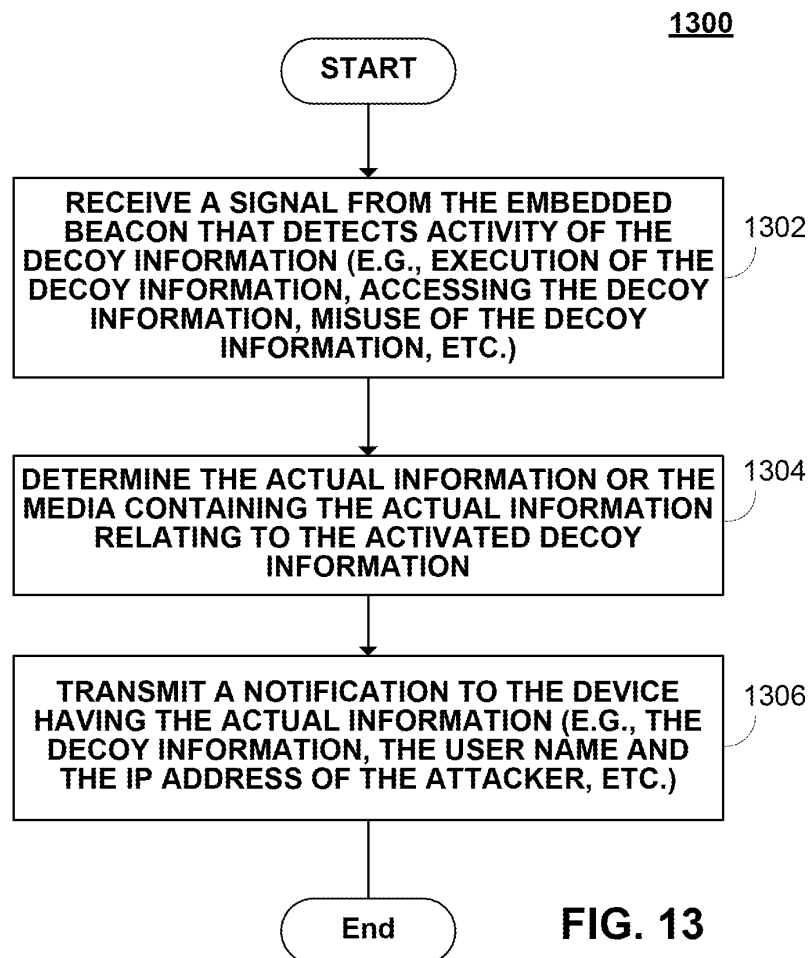
	TIME	IP	BODY
1	Mon Jun 9 14:54:10 2008	192.160.1.100	document=pdf pass=9c92ce63e9432b6f2c3c958d6437e842 title=Taxes os=WIN version=8.1
2	Mon Jun 9 14:53:52 2008	192.160.1.100	document=pdf pass=9c92ce63e9432b6f2c3c958d6437e842 title=Taxes os=WIN version=8.1
3	Mon Jun 9 14:49:33 2008	192.160.1.100	document=pdf pass=9c92ce63e9432b6f2c3c958d6437e842 title=Taxes os=WIN version=8.1
4	Mon Jun 9 14:30:20 2008	192.160.1.100	document=pdf pass=9c92ce63e9432b6f2c3c958d6437e842 title=Taxes os=WIN version=7
5	Fri Jun 6 10:54:21 2008	209.2.1.100	document=pdf pass=9c92ce63e9432b6f2c3c958d6437e842 title=Taxes os=WIN version=8.1
6	Thu Jun 5 09:29:50 2008	69.116.89.159	document=pdf pass=9c92ce63e9432b6f2c3c958d6437e842 title=Taxes os=WIN version=8.1
7	Thu Jun 5	69.116.89.159	document=pdf pass=073dad43af7048b666c8655e51734a97 title=Sh...

FIG. 11

12/13

**FIG. 12**

13/13



INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2008/066623

A. CLASSIFICATION OF SUBJECT MATTER IPC(8) - G06F 12/14 (2009.01) USPC - 726/24 According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) IPC(8) - G06F 12/14 (2009.01) USPC - 726/24 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) MicroPatent		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2007/0101430 A1 (RAIKAR) 03 May 2007 (03.05.2007) entire document	1-2,4-28,30-52
Y		3,29
Y	US 7,093,291 B2 (BAILEY) 15 August 2006 (15.08.2006) entire document	3-29
A	US 2004/0172557 A1 (NAKAE et al) 02 September 2004 (02.09.2004) entire document	1-52
☐ Further documents are listed in the continuation of Box C. ☐		
* 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 04 February 2009		Date of mailing of the international search report 13 FEB 2009
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201		Authorized officer: Blaine R. Copenheaver PCT Helpdesk: 571-272-4300 PCT QSP: 571-272-7774

Form PCT/ISA/210 (second sheet) (April 2005)