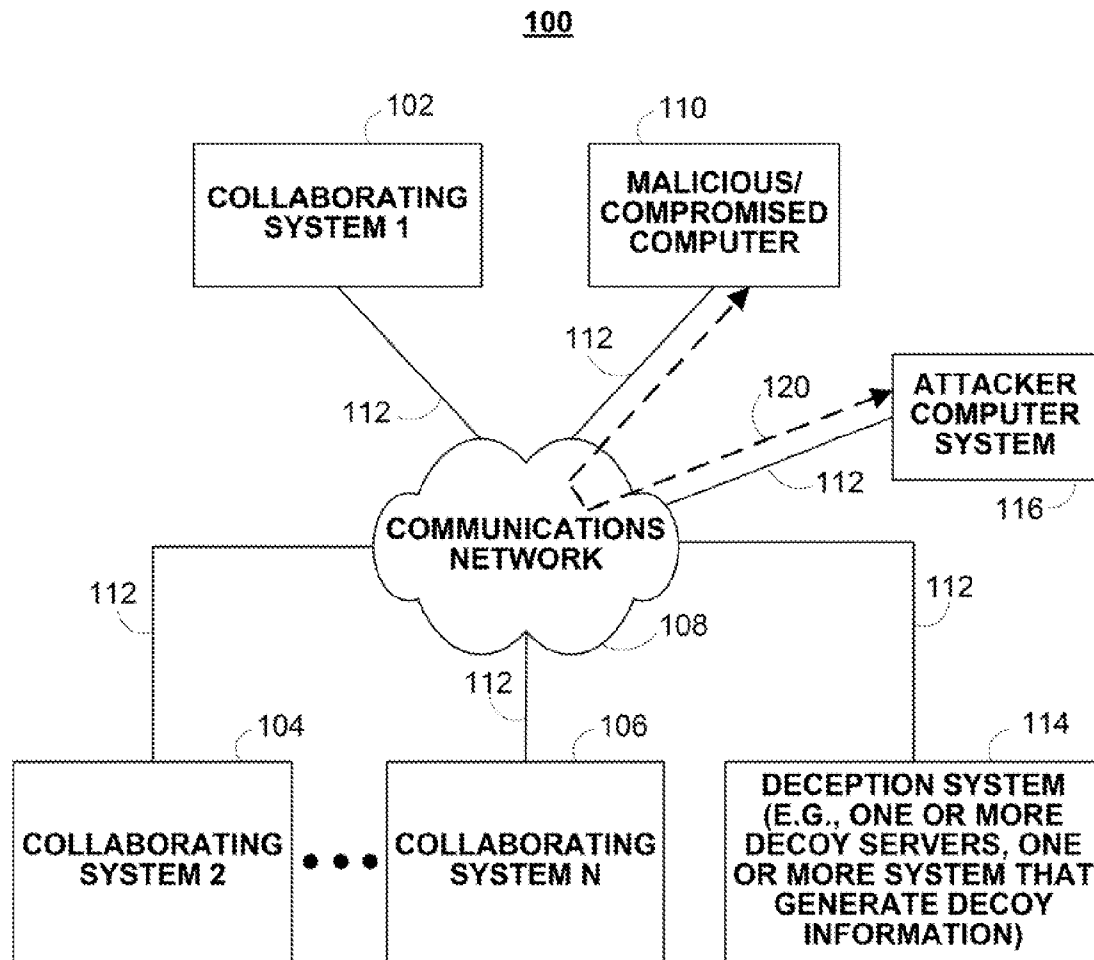




US 20120084866A1

(19) **United States**(12) **Patent Application Publication**
Stolfo(10) **Pub. No.: US 2012/0084866 A1**(43) **Pub. Date: Apr. 5, 2012**(54) **METHODS, SYSTEMS, AND MEDIA FOR
MEASURING COMPUTER SECURITY**cation No. 61/099,526, filed on Sep. 23, 2008, provi-
sional application No. 61/165,634, filed on Apr. 1,
2009.(76) Inventor: **Salvatore J. Stolfo**, Ridgewood, NJ
(US)**Publication Classification**(21) Appl. No.: **13/166,723**(51) **Int. Cl.**
G06F 21/00 (2006.01)(22) Filed: **Jun. 22, 2011**(52) **U.S. Cl.** **726/25****Related U.S. Application Data**(57) **ABSTRACT**(63) Continuation-in-part of application No. 12/565,394,
filed on Sep. 23, 2009, which is a continuation-in-part
of application No. PCT/US2008/066623, filed on Jun.
12, 2008.(60) Provisional application No. 61/357,481, filed on Jun.
22, 2010, provisional application No. 60/934,307,
filed on Jun. 12, 2007, provisional application No.
61/044,376, filed on Apr. 11, 2008, provisional appli-cation No. 61/099,526, filed on Sep. 23, 2008, provi-
sional application No. 61/165,634, filed on Apr. 1,
2009.
Methods, systems, and media for measuring computer secu-
rity are provided. In accordance with some embodiments,
methods for measuring computer security are provided, the
methods comprising: making at least one of decoys and non-
threatening access violations accessible to a first user using a
computer programmed to do so; maintaining statistics on
security violations and non-violations of the first user using a
computer programmed to do so; and presenting the statistics
on a display.

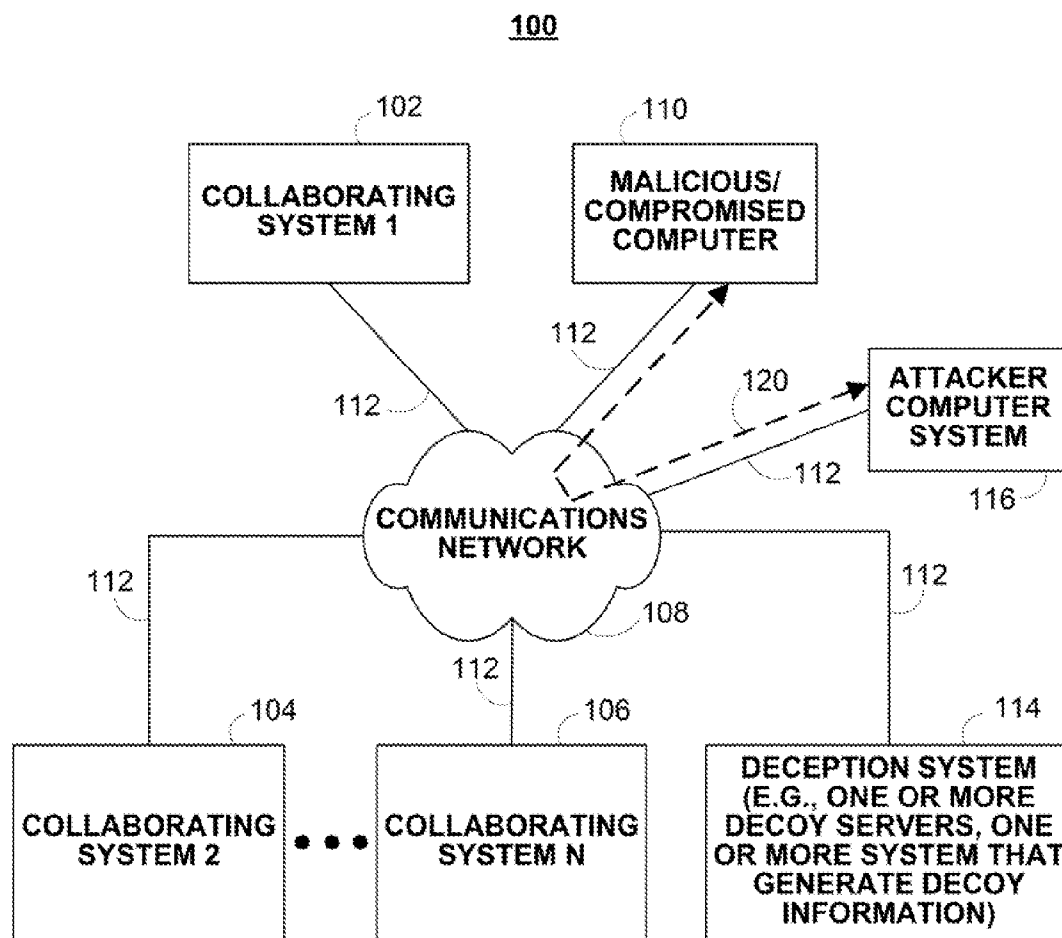


FIG. 1

200

202

testtax.pdf - Adobe Acrobat Professional

File Edit View Document Comments Forms Tools Advanced Window

Create PDF Combine Files Export Secure Sign

1/2 73.9% Find

Form 1040 Department of the Treasury - Internal Revenue Service 2007 IRS use only - Do not

U.S. Individual Income Tax Return

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

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

YOUR LABEL HERE

Your first name and Initial Barbara Last name Denison

If a joint return, spouse's first name and Initial Last name

Home address (number and street). If you have a P.O. box, see page 12 4293 Red Hawk Road

City, town or post office, state, and ZIP code. If you have a foreign address, Long Prairie, MN 56347

Check here if you, or your spouse if filing jointly, want \$3 to go to this fund

204

testtax.pdf - Adobe Acrobat Professional

File Edit View Document Comments Forms Tools Advanced Window

Create PDF Combine Files Export Secure Sign

1/2 73.9% Find

Form 1040 Department of the Treasury - Internal Revenue Service 2007 IRS use only - Do not

U.S. Individual Income Tax Return

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

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

YOUR LABEL HERE

Your first name and Initial Barbara Last name Denison

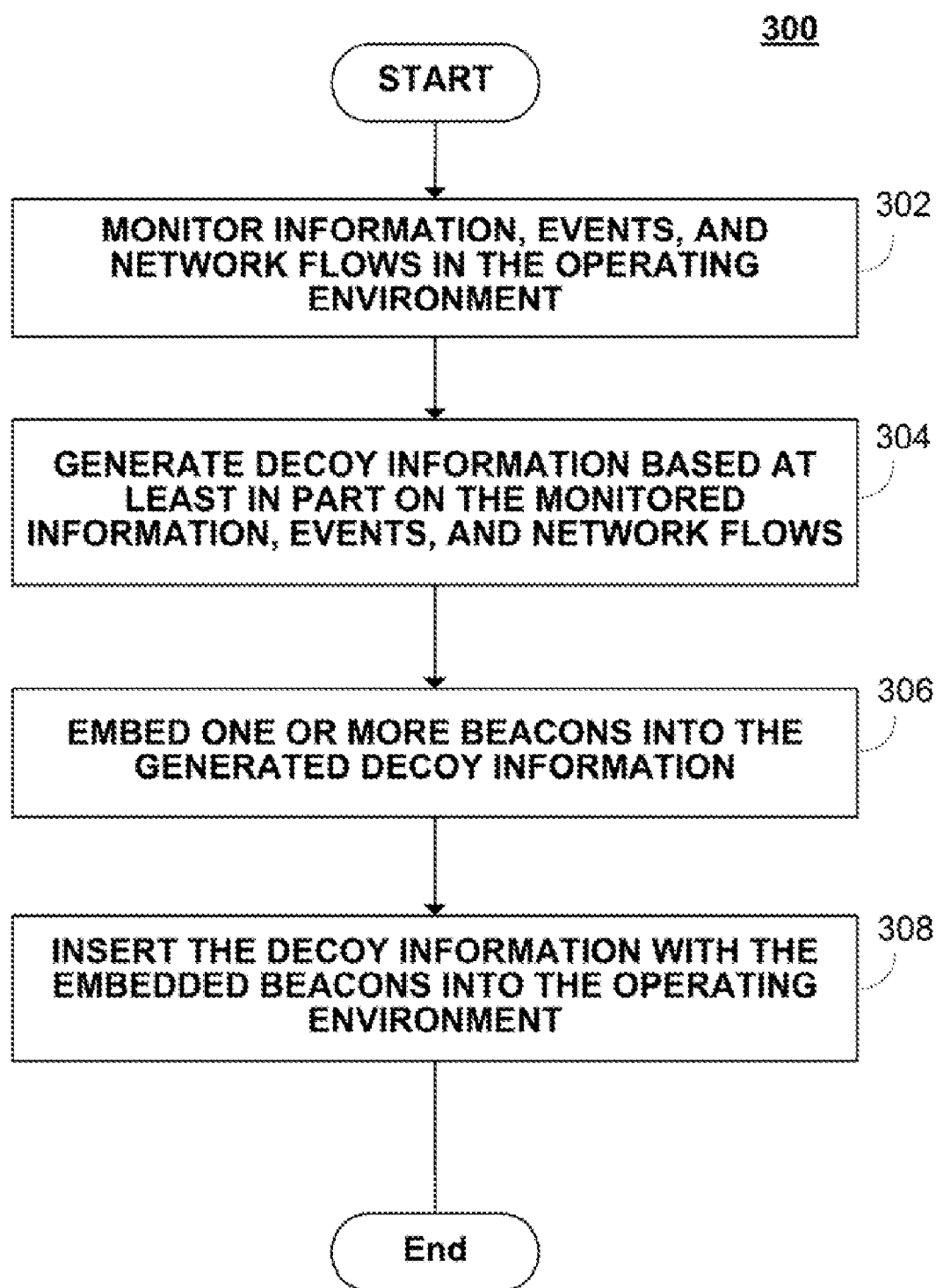
If a joint return, spouse's first name and Initial Last name

Home address (number and street). If you have a P.O. box, see page 12 4293 Red Hawk Road

City, town or post office, state, and ZIP code. If you have a foreign address, Long Prairie, MN 56347

Check here if you, or your spouse if filing jointly, want \$3 to go to this fund

FIG. 2

**FIG. 3**

400			
Time	Source	Destination	Protocol Info
2.079658	204.127.135.135	51.135.135.135	SMTP Response: 220 comcast.net
2.080211	160.39.251.135	204.127.192.17	SMTP Command: EHLO [160.39.251.135]
2.165540	204.127.192.17	160.39.251.135	SMTP Response: 250 - comcast.net
2.166357	160.39.251.135	204.127.192.17	SMTP Command: AUTHPLAINAUTHRC3QxMmFjY291bnQx
2.606035	160.39.251.135	204.127.192.17	SMTP Command: RCPT TO: <bmbowen@gmail.com>
MACs			
> Frame 197 (125 bytes on wire, 125 bytes captured) > Ethernet II, Src: Cisco_26:9c:00 (00:d0:06:26:9c:00), Dst: AppleCom_4c:d8:a4 (00:17:f2:4c:d8:a4) > Destination: AppleCom_4c:d8:a4 (00:17:f2:4c:d8:a4) > Source: Cisco_26:9c:00 (00:d0:06:26:9c:00)			
402			
Time	Source	Destination	Add an expression to this filter string
8.411955	63.240.76.10	160.39.247.72	POP Response: +OK (scrrpx)
8.412001	160.39.247.72	63.240.76.10	TCP 45191 > pop3 [ACK] Seq=1234
8.412019	160.39.247.72	63.240.76.10	POP Request: CAPA
8.462934	160.39.247.72	63.240.76.10	POP Request: USER test12account1234
8.468745	160.39.247.72	63.240.76.10	POP Request: PASS wxyz9876
Password			
> Frame 30 (118 bytes on wire, 118 bytes captured) > Ethernet II, Src: Cisco_26:9c:00 (00:d0:06:26:9c:00), Dst: AppleCom_4c:d8:a4 (00:17:f2:4c:d8:a4) > Destination: AppleCom_4c:d8:a4 (00:17:f2:4c:d8:a4) > Source: Cisco_26:9c:00 (00:d0:06:26:9c:00)			

FIG. 4

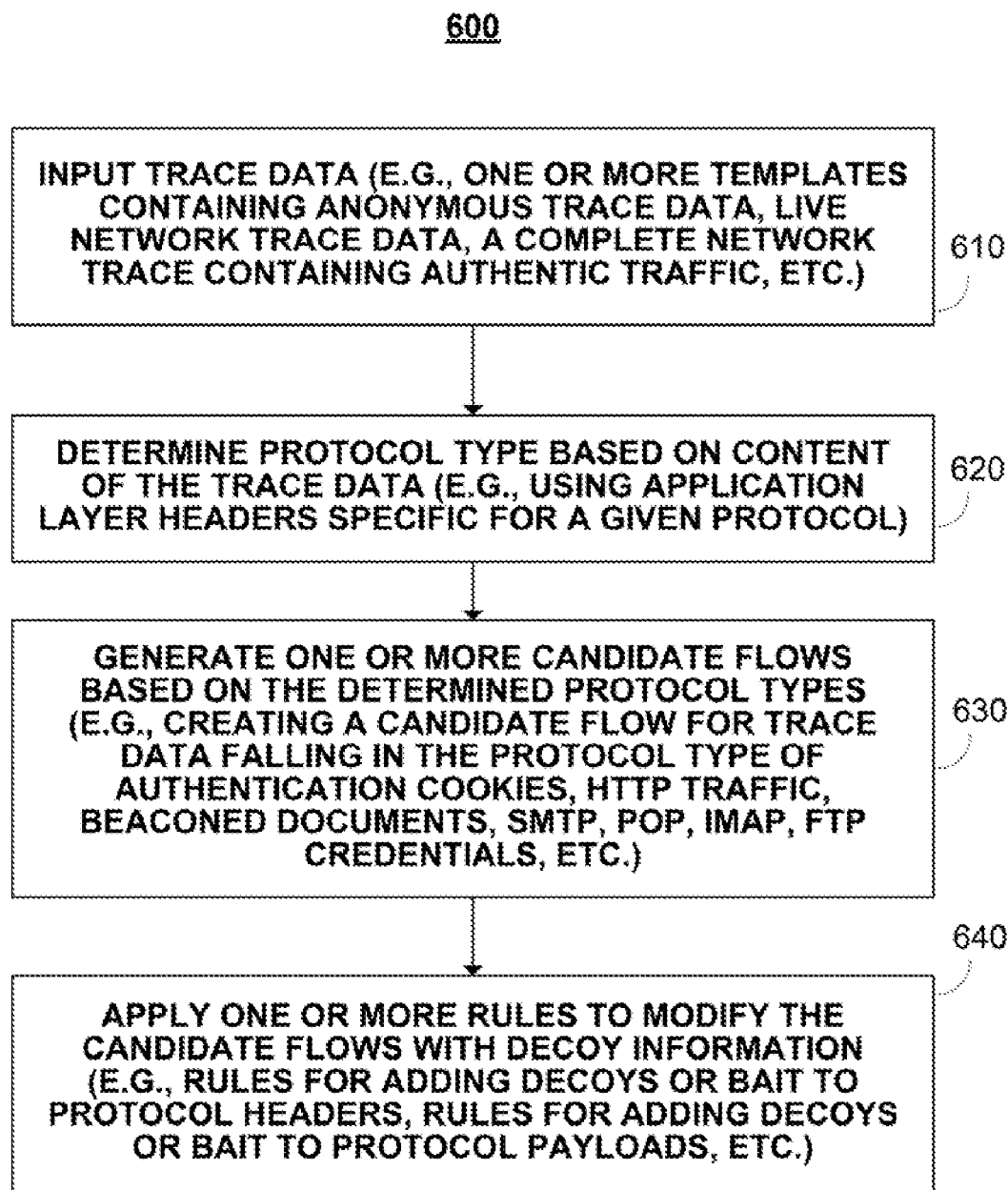
500

Time	Source	Destination	Protocol	Info
2.152072	Modified IPs for testing	1	TCP	38401 > smtp [ACK] S
2.198952	1.1.1.1	0.0.0.0	SMTP	Response: 220 cs1.com
2.199585	0.0.0.0	1.1.1.1	SMTP	Modified protocol commands to remove actual SMTP server
2.254768	0.0.0.0	1.1.1.1	SMTP	Response: 220 cs1.com
2.379277	1.1.1.1	0.0.0.0	SMTP	Response: 220 cs1.com
Bogus MACs				
Frame 29 (74 bytes on wire, 74 bytes captured)				
Ethernet II, Src: b0:92:d5:0b:c4:77 (b0:92:d5:0b:c4:77), Dst: 36:f2:89:d4:f0:73 (Dst: 36:f2:89:d4:f0:73)				
Destination: 36:f2:89:d4:f0:73 (Dst: 36:f2:89:d4:f0:73)				
Source: b0:92:d5:0b:c4:77 (b0:92:d5:0b:c4:77)				

502

Time	Source	Arbitrary wireless CU IP as source	Destination	Protocol	Info
6.171231	128.59.23.85	160.39.247.72	160.39.247.72	POP	Response: +OK
6.222646	128.59.23.85	160.39.247.72	160.39.247.72	POP	Response: +OK Capability list follows
6.222992	160.39.247.72	128.59.23.85	128.59.23.85	POP	Response: USER decoyuser1234
6.228726	128.59.23.85	160.39.247.72	160.39.247.72	TCP	pop3 > 45191 [ACK] Seq+250 Ack+27 Win+33304
6.228753	160.39.247.72	160.39.247.72	160.39.247.72	POP	Request: PASS decoypassword
Bogus MAC resembling a wireless linksys NIC					
Frame 1014 (74 bytes on wire, 74 bytes captured)					
Ethernet II, Src: LinksysG_d4:f0:73 (00:06:25:d4:f0:73), Dst: b0:92:d5:0b:c4:77					
Destination: b0:92:d5:0b:c4:77 (b0:92:d5:0b:c4:77)					
Source: LinksysG_d4:f0:73 (00:06:25:d4:f0:73)					

FIG. 5

**FIG. 6**

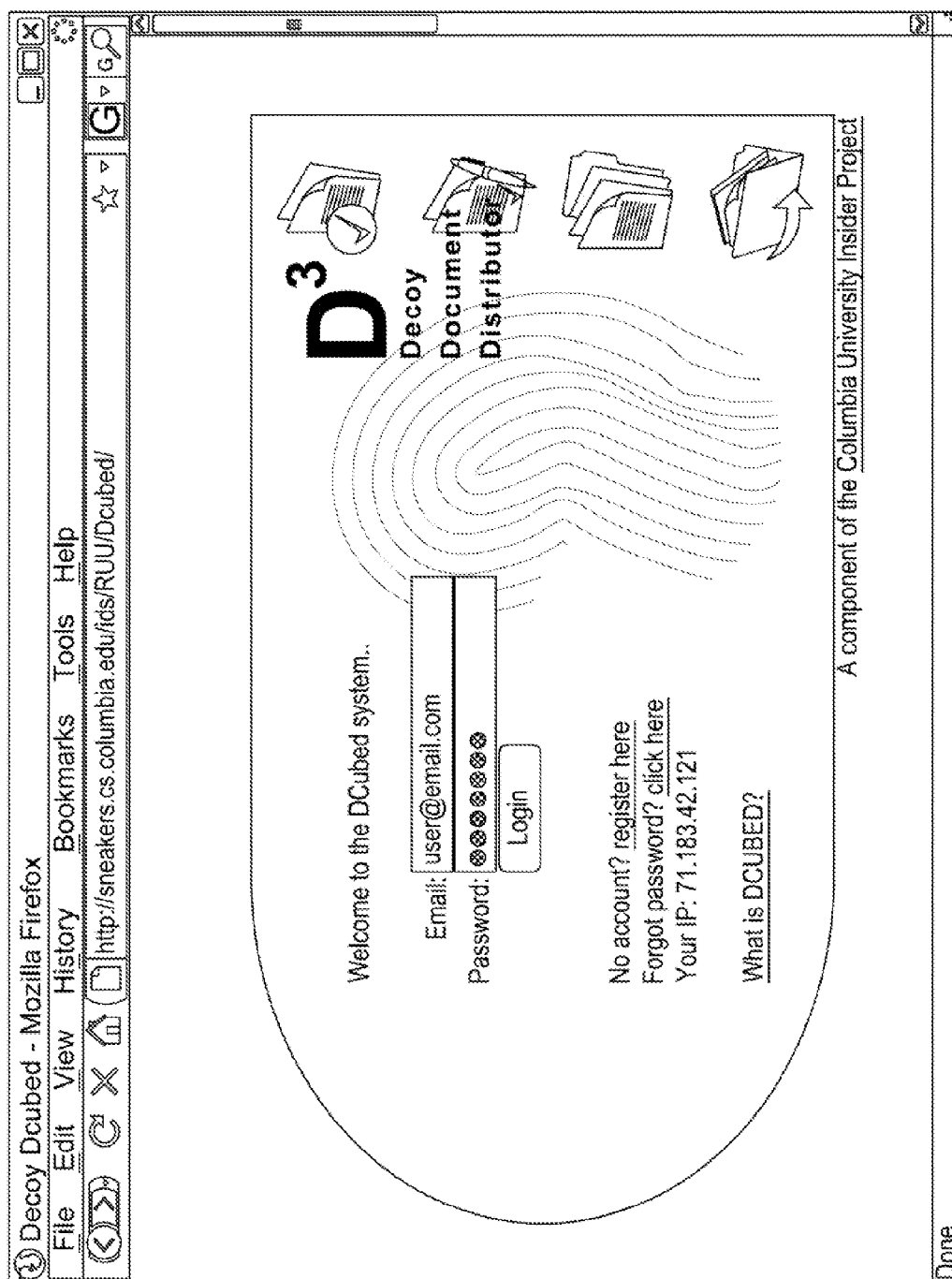


FIG. 7

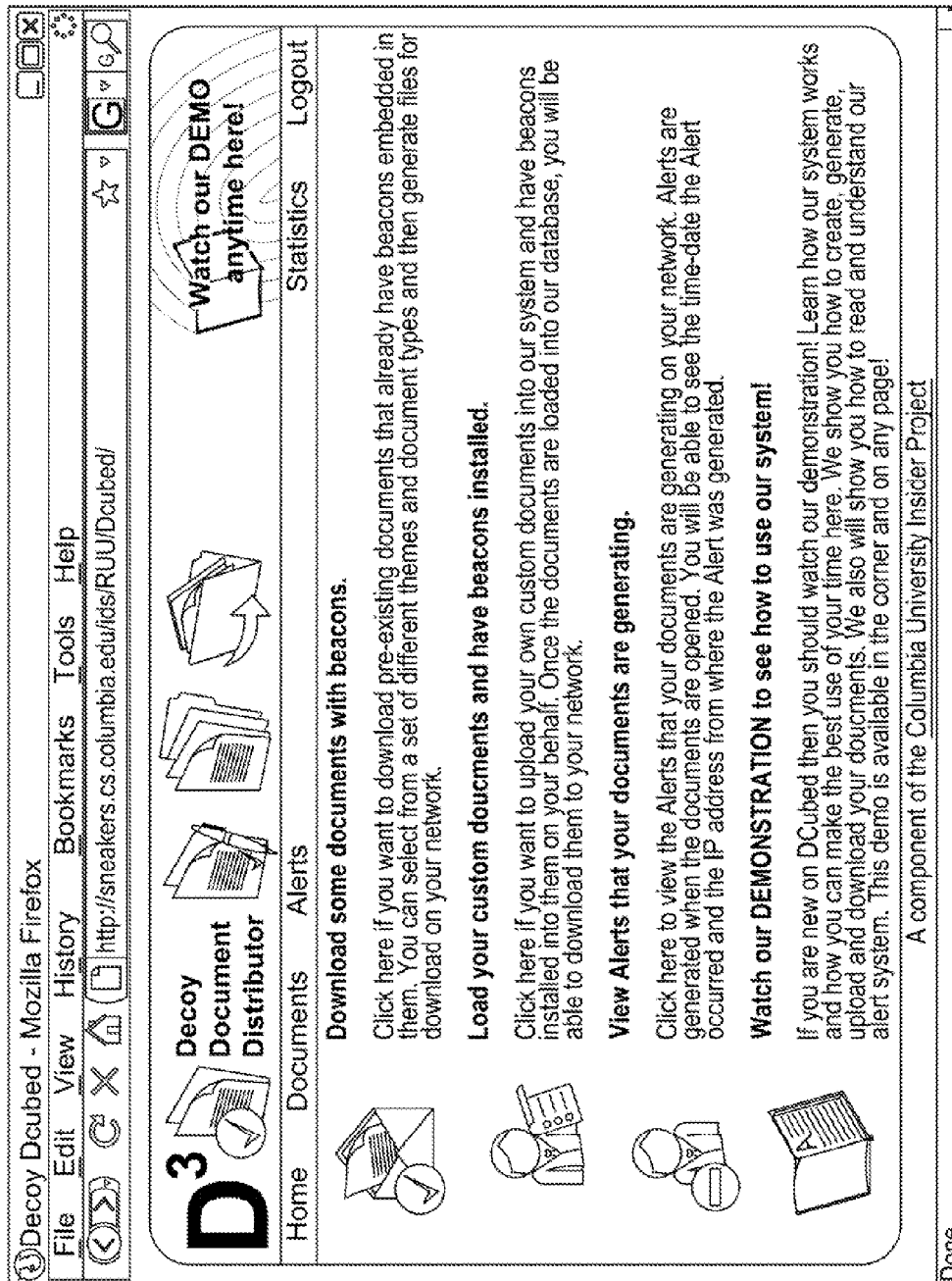


FIG. 8

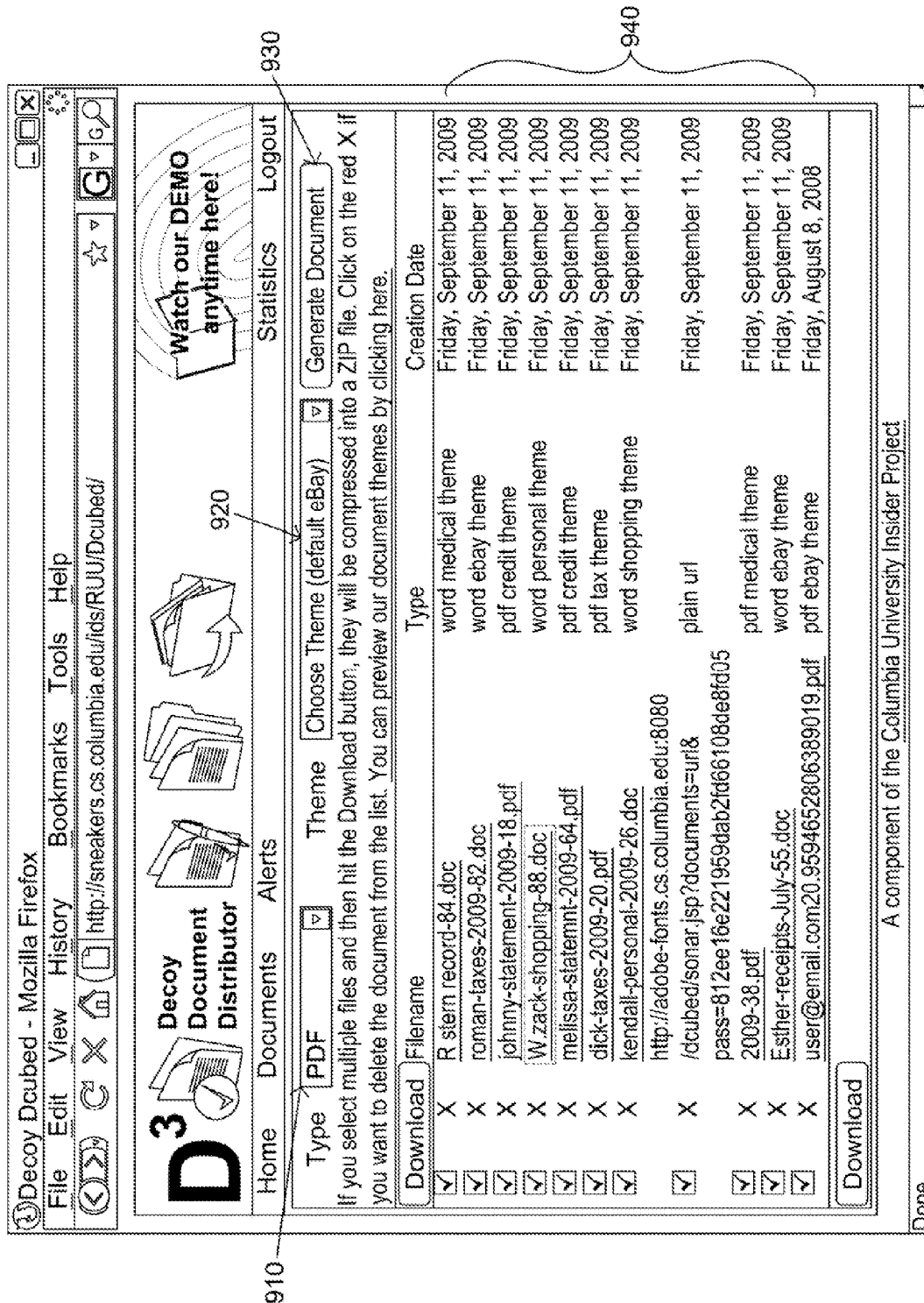


FIG. 9

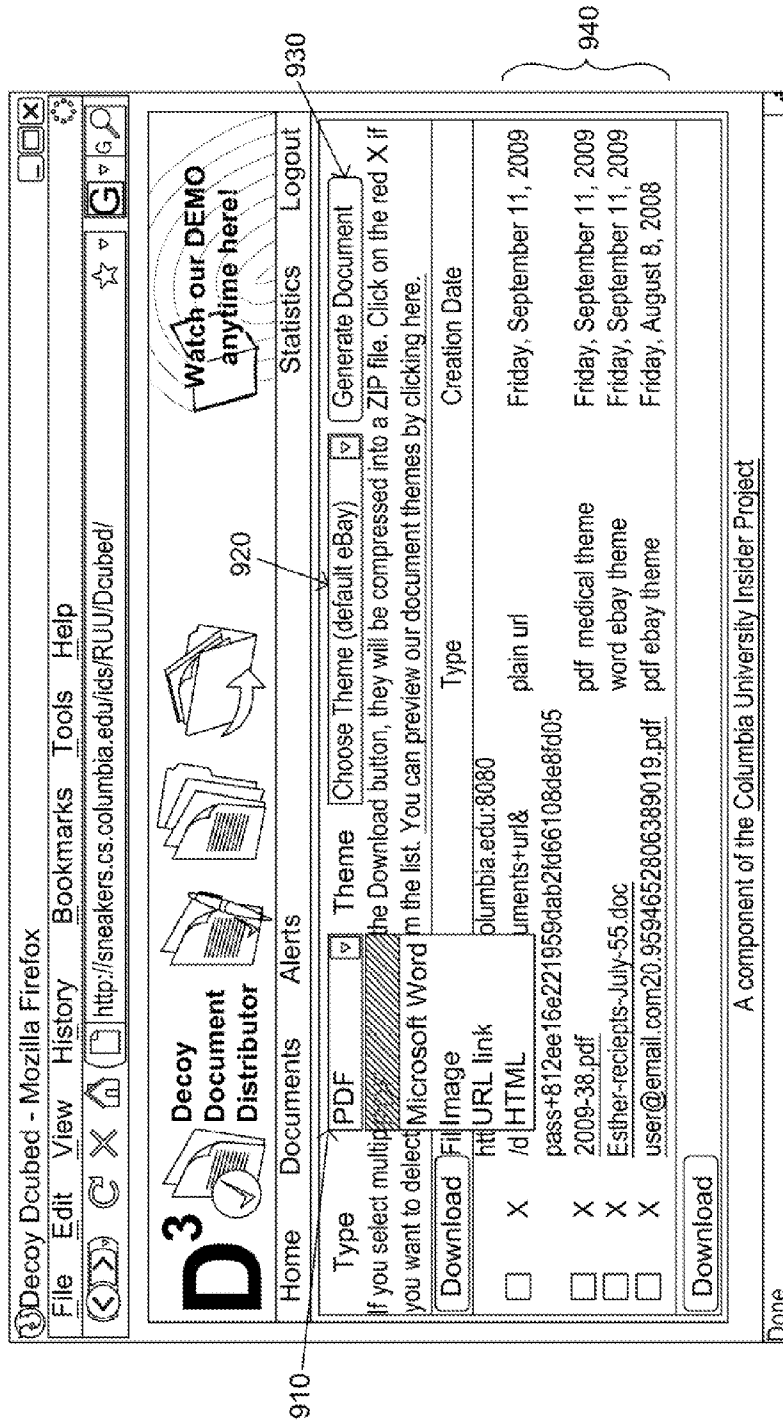


FIG. 10

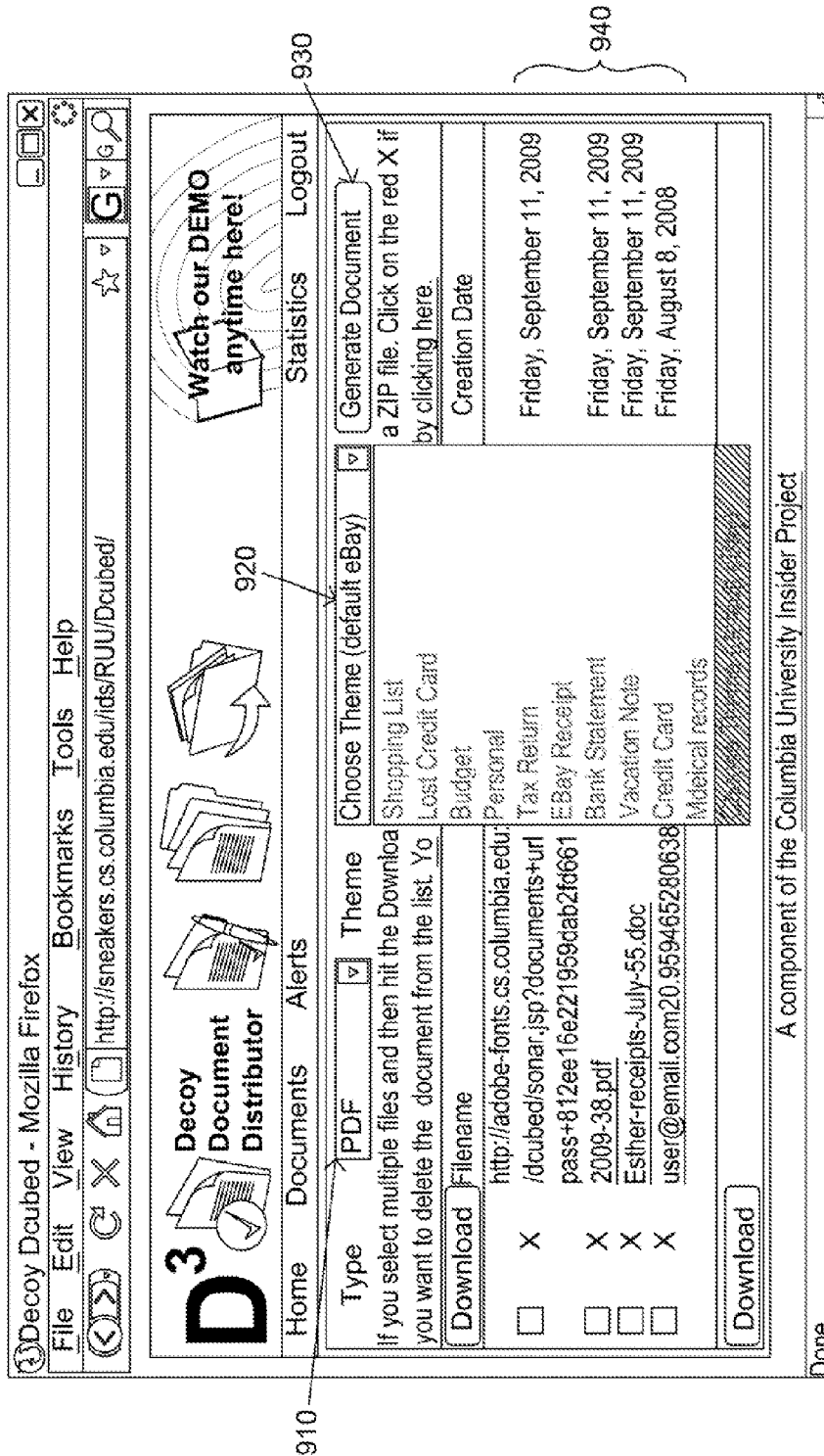


FIG. 11

										Esther-receipts-July-5.doc(Read-Only) [Compatibility Mode] - Microsoft Word											
Home		Insert		PageLayout		References		Mailings		Review		Developer		Add-Ins		Acrobat					
 Clipboard Paste		Times New Roman ▼ 12 ▼		B / U ▾ abc x ² Aa ▾												AA Change Styles ▾ ¶ Normal No Spad Heading 1 ▾ Styles					
Font		Paragraph																			
Security Warning Macros have been disabled. Options...																					
eBay®																					
Order Summary																					
computer mouse																					
Logitech VX Nano Cordless Laser Mouse for Notebooks																					
dvd																					
The Great Debaters																					
Price: \$50.99																					
Price: \$18.99																					
Sub:\$69.98																					
Shipping: \$5.11																					
Total: \$75.09																					
<table border="1"> <tr> <th>Ship To</th> <th>Bill To</th> </tr> <tr> <td>Edna Morris 4739 Comfort Court Madison, WI 53703</td> <td>Visa:4929809195625938 Expires: 1/2012 Email: Edna.J.Morris@dodgit.com</td> </tr> </table>																		Ship To	Bill To	Edna Morris 4739 Comfort Court Madison, WI 53703	Visa:4929809195625938 Expires: 1/2012 Email: Edna.J.Morris@dodgit.com
Ship To	Bill To																				
Edna Morris 4739 Comfort Court Madison, WI 53703	Visa:4929809195625938 Expires: 1/2012 Email: Edna.J.Morris@dodgit.com																				
BhJvy30fA																					
Place my order!																					
Return confirmation number: BRtEzQLIZ(in case you haven't already received this form)																					

FIG. 12

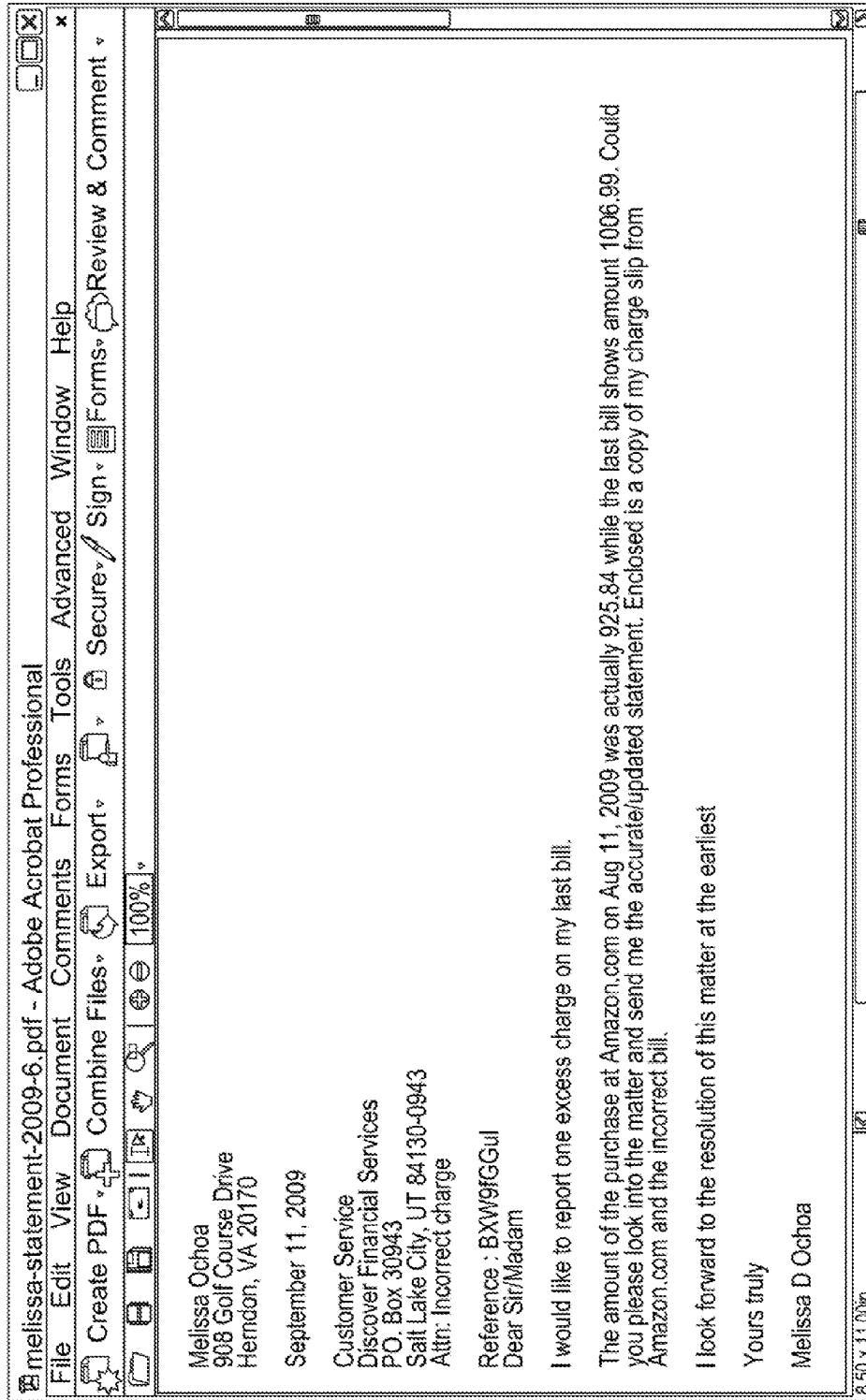


FIG. 13

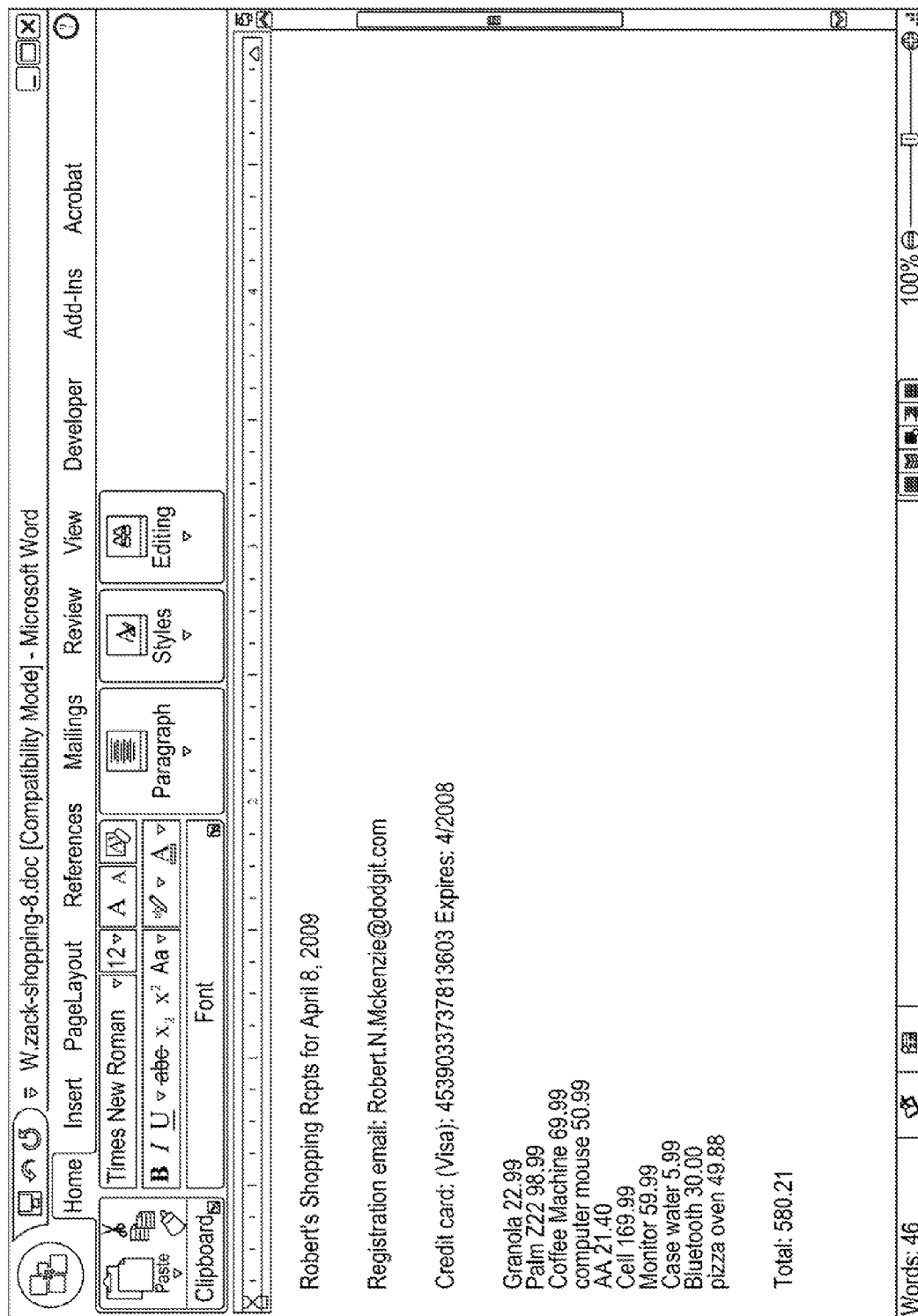


FIG. 14

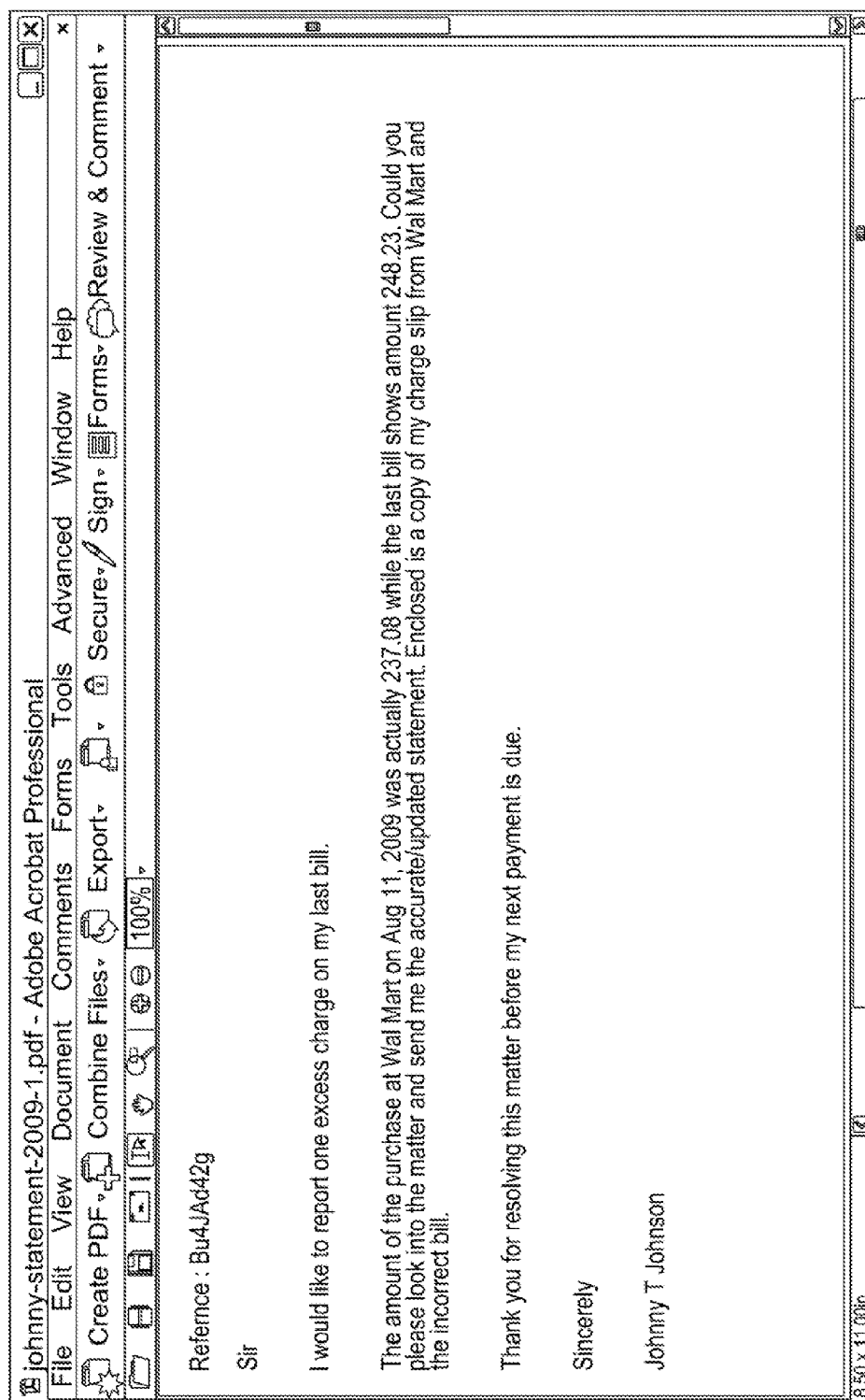


FIG. 15

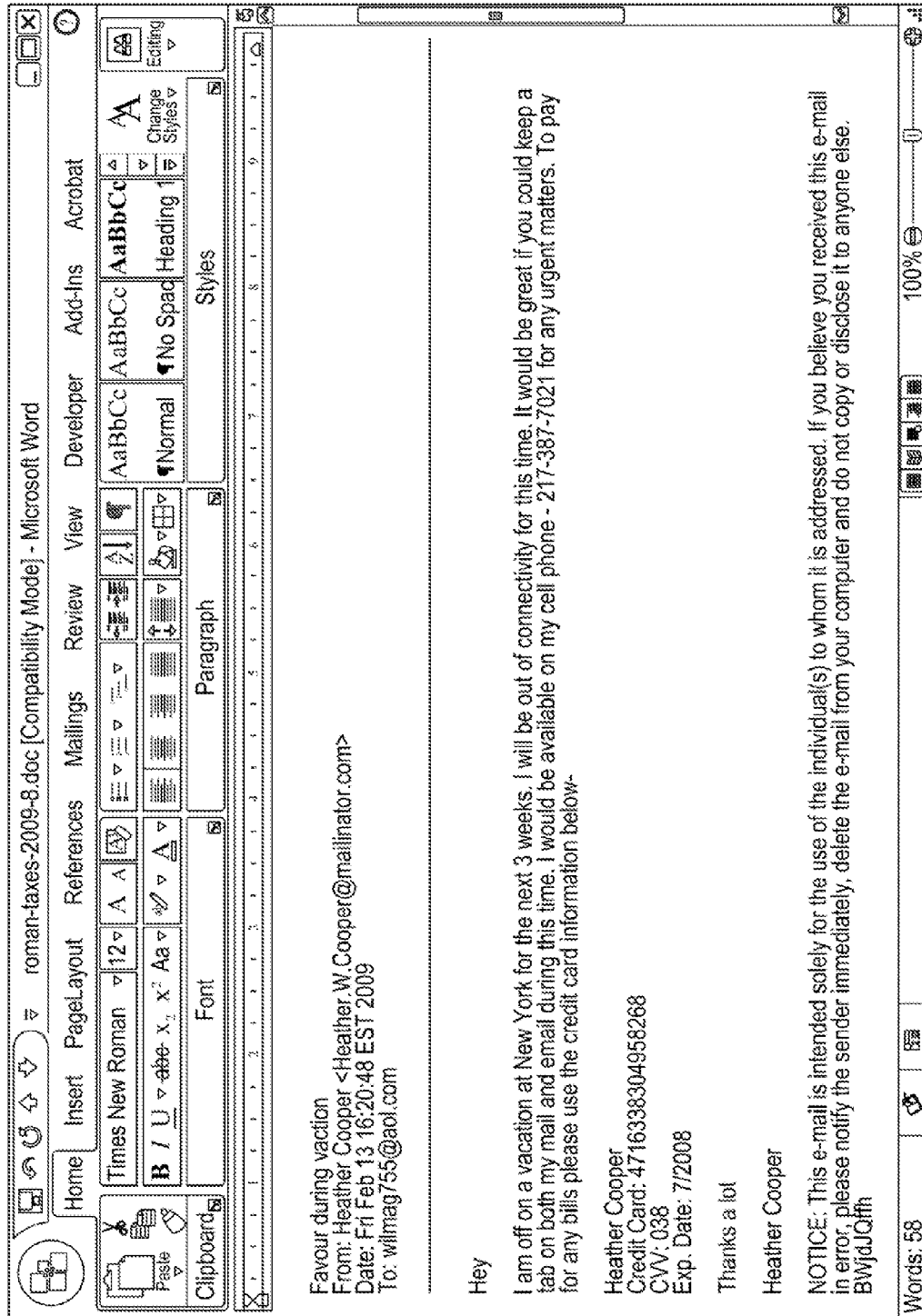


FIG. 16

Date	Description	Amount
April 8, 2009	Emergency Treatment Costs	1587.55
	Prescription	498.26
		2085.82

Signed by: Marta Markowski

FIG. 17

Form 1040 Department of the Treasury - Internal Revenue Service 2007 IRS use only - Do not

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

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

Your first name and Initial Last name
Dick Kimball

If a joint return, spouse's first name and Initial Last name

Home address (number and street). If you have a P.O. box, see page 12
838 Cost Avenue

City, town or post office, state, and ZIP code. If you have a foreign address,
Beltsville, MD 20705

Presidential Election Campaign Check here if you, or your spouse if filing jointly, want \$3 to go to this fund

Filing Status

1 ☒ Single 4 ☐ Head of household

2 ☐ Married filing jointly (even if only one had income) the qualifying

3 ☐ Married filing separately. Enter spouse's SSN above this child's na

5 ☐ Qualifying widow

Exemptions

6a ☒ Yourself. If 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

If more than four dependents, see page 15.

d Total number of exemptions claimed - - - - -

Income

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

8a Taxable interest. Attach Schedule B if required - - - - -

b Tax-exempt interest. Do not include on line 8a - - - - -

9a Ordinary dividends. Attach Schedule B if required - - - - -

b Qualified dividends (see page 19) - - - - -

10 Taxable refunds, credits, or offsets of state and local income taxes (see page 2)

11 Alimony received - - - - -

12 Business income or (loss). Attach Schedule C or C-EZ - - - - -

13 Capital gain or (loss). Attach Schedule D if required, check here - - - - -

14 Other gains or (losses). Attach Form 4797 - - - - -

15a IRA distributions - - - - - 15a b Taxable amount (see

FIG. 18

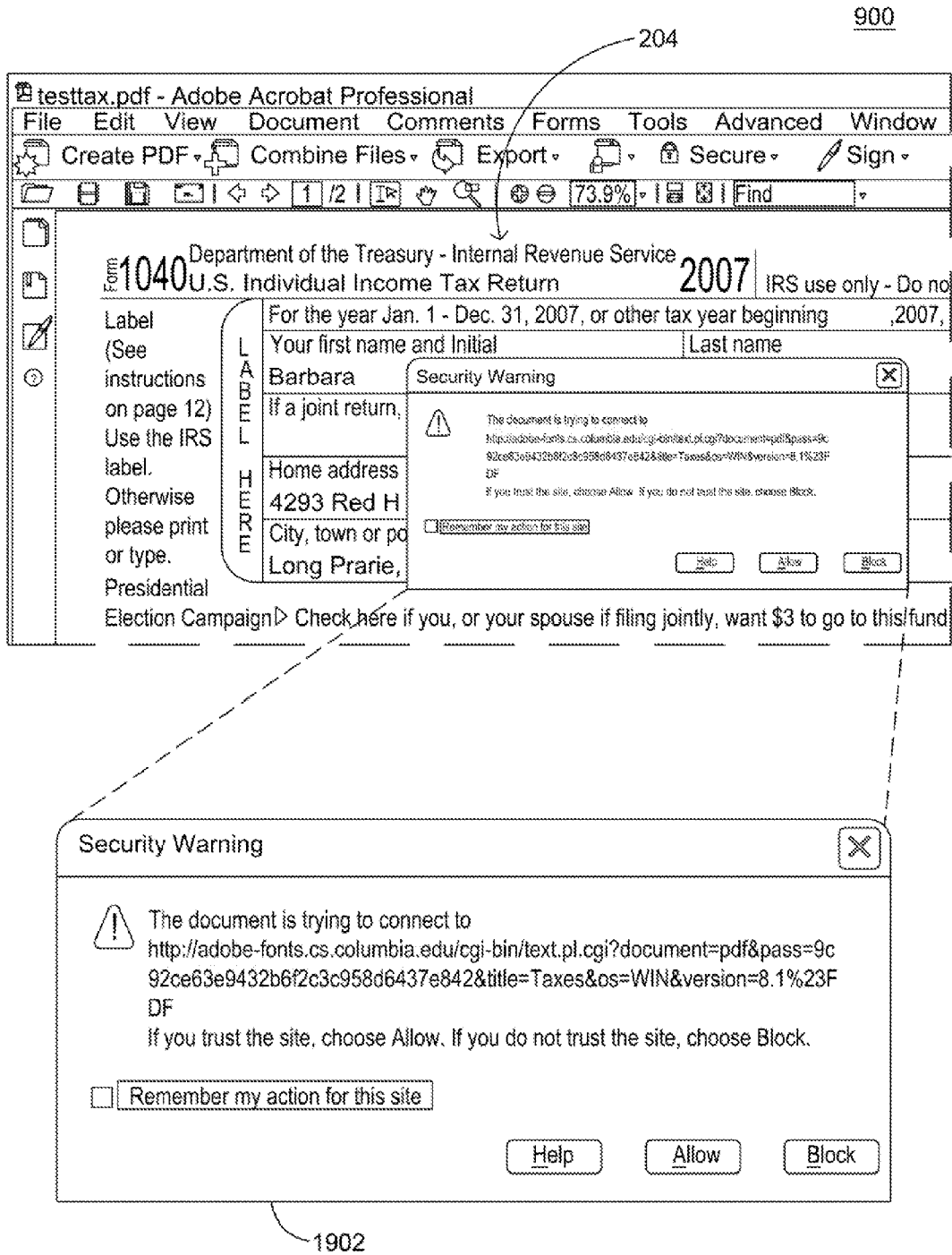


FIG. 19

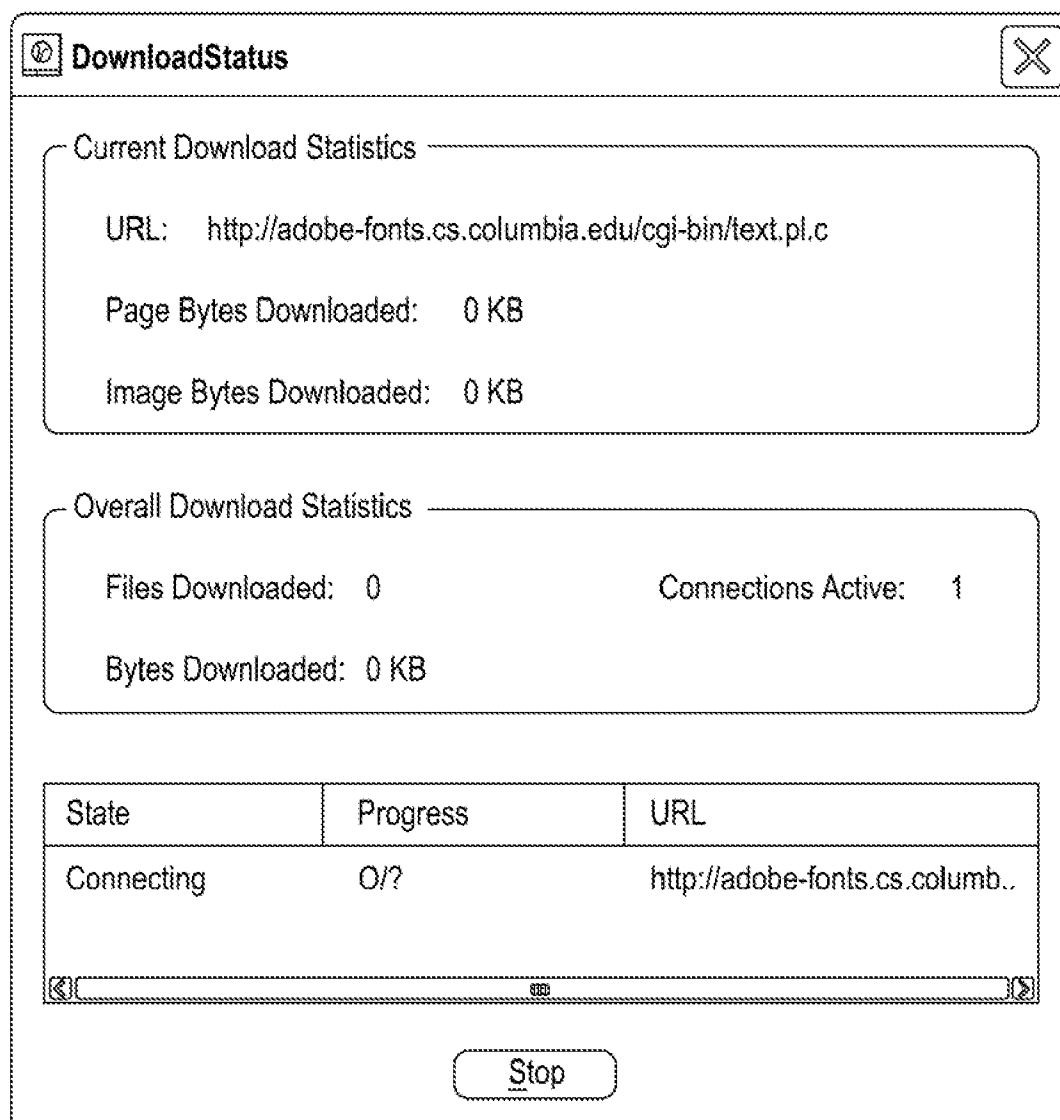


FIG. 20

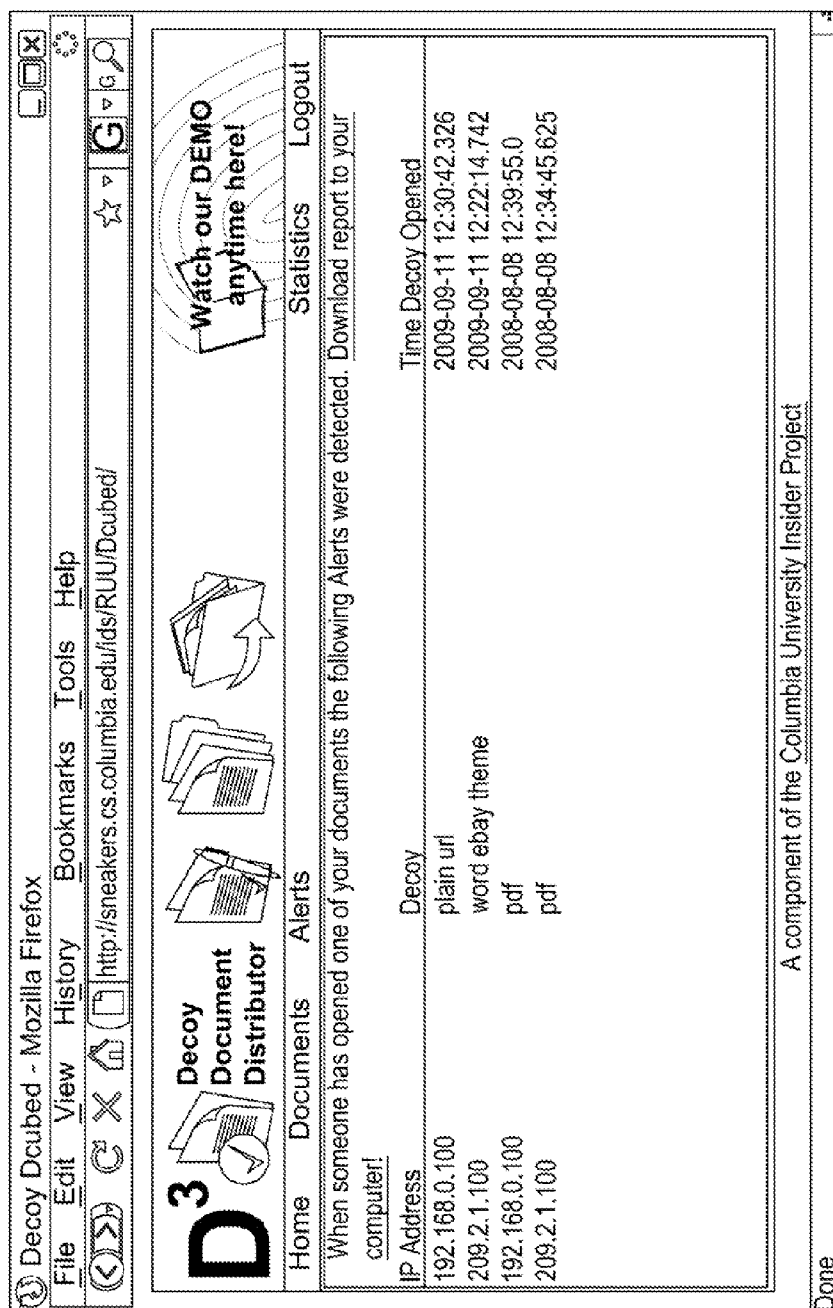


FIG. 21

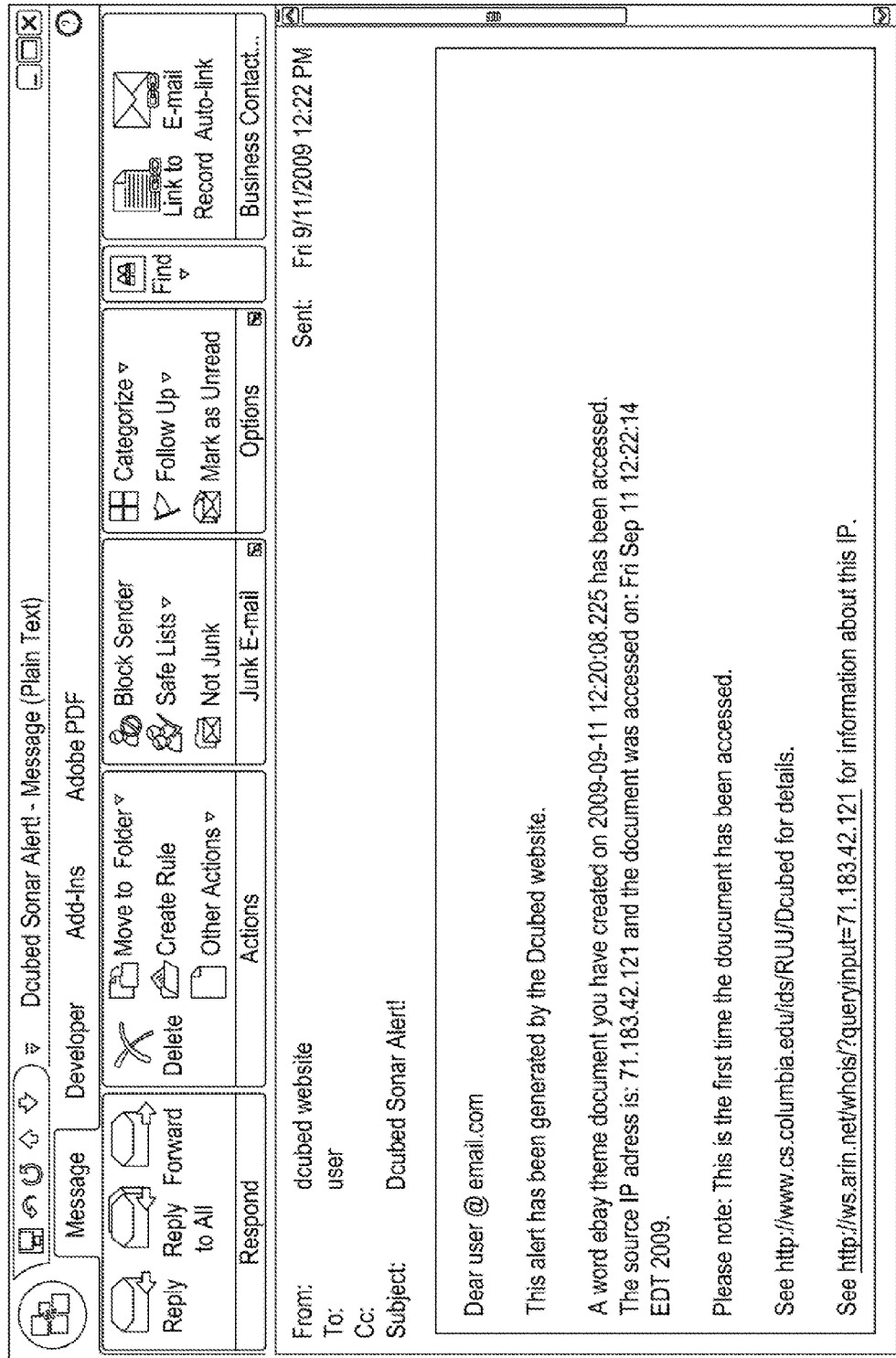
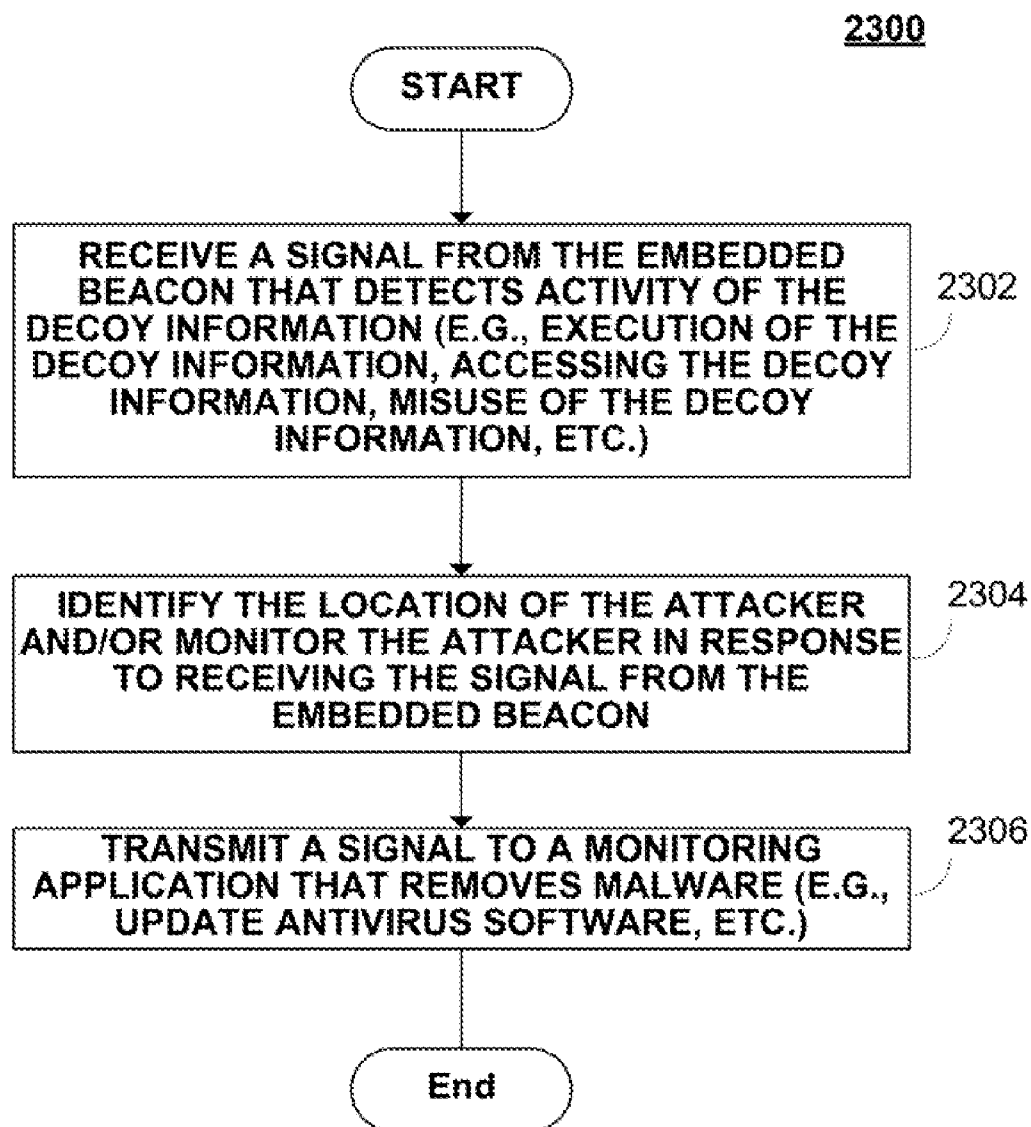
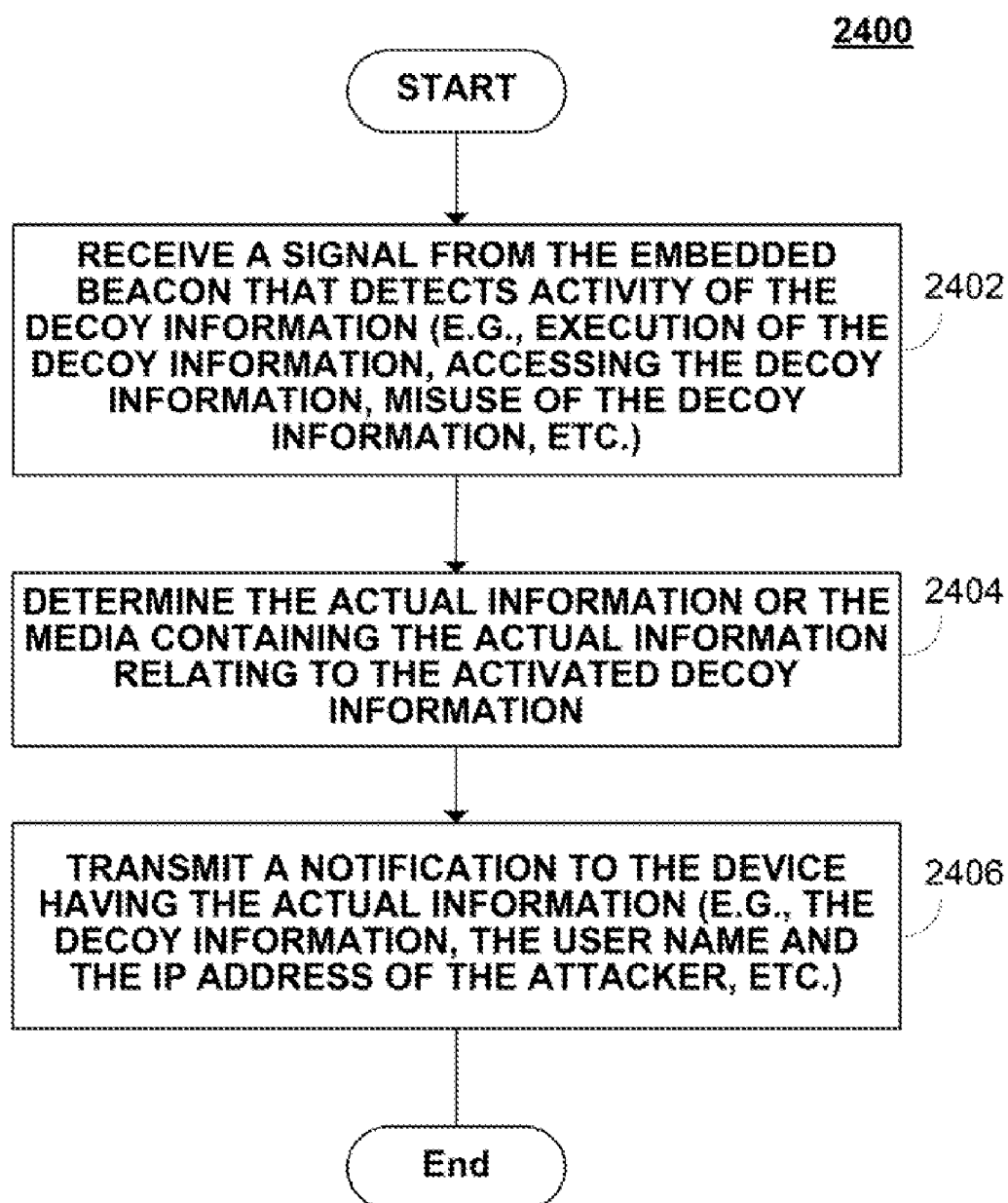
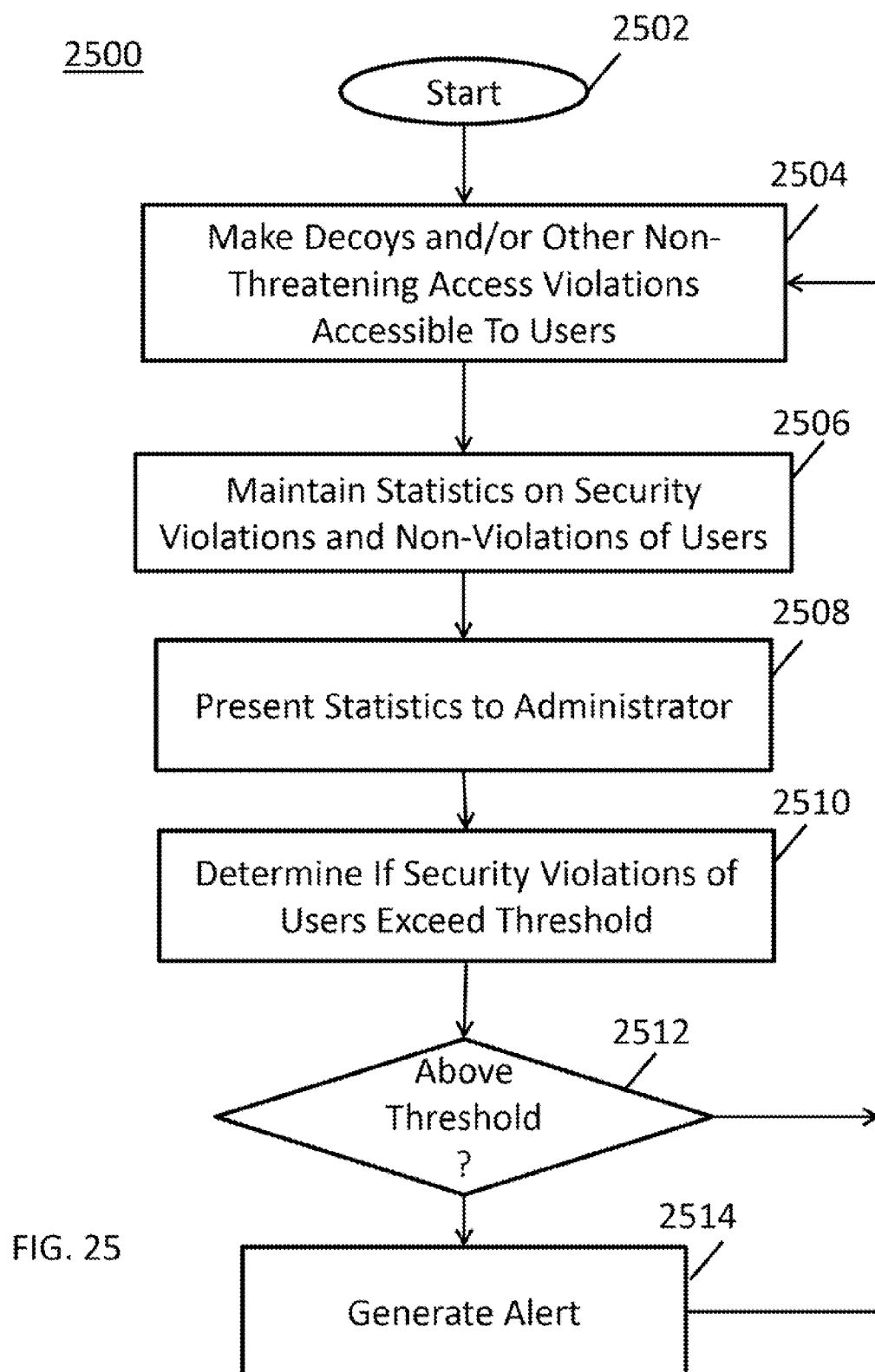


FIG. 22

**FIG. 23**

**FIG. 24**



METHODS, SYSTEMS, AND MEDIA FOR MEASURING COMPUTER SECURITY

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Provisional Patent Application No. 61/357,481, filed Jun. 22, 2010, and is a continuation-in-part of U.S. patent application Ser. No. 12/565,394, filed Sep. 23, 2009, each of which is hereby incorporated by reference herein in its entirety.

[0002] U.S. patent application Ser. No. 12/565,394, filed Sep. 23, 2009, is a continuation-in-part of International Application No. PCT/US2008/066623, filed Jun. 12, 2008, which claims the benefit of U.S. Provisional Patent Application No. 60/934,307, filed Jun. 12, 2007 and U.S. Provisional Patent Application No. 61/044,376, filed Apr. 11, 2008, which are hereby incorporated by reference herein in their entireties. U.S. patent application Ser. No. 12/565,394, filed Sep. 23, 2009, also claims the benefit of U.S. Provisional Patent Application No. 61/099,526, filed Sep. 23, 2008 and U.S. Provisional Application No. 61/165,634, filed Apr. 1, 2009, which are hereby incorporated by reference herein in their entireties.

[0003] U.S. patent application Ser. No. 12/565,394, filed Sep. 23, 2009, 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

[0004] The invention was made with government support under Grant No. 60NANBID0127 awarded by the U.S. Department of Homeland Security through the Institute for Information Infrastructure Protection (I3P), under Grant No. W911NF-06-1-0151-49626-CI awarded by the Army Research Office (ARO), and under Grant No. CNS-07-14647 awarded by the National Science Foundation (NSF). The government has certain rights in the invention.

TECHNICAL FIELD

[0005] The disclosed subject matter relates to methods, systems, and media for measuring computer security.

BACKGROUND

[0006] Much research in computer security has focused on approaches for preventing unauthorized and illegitimate access to systems and information. However, one of the most damaging malicious activities is the result of internal misuse within an organization. This may be because much of the attention has been focused on preventative measures against computer viruses, worms, trojans, hackers, rootkits, spyware, key recovery attacks, denial-of-service attacks, malicious software (or malware), probes, etc. such that far less attention has been focused inward.

[0007] Insider threats generally include masqueraders and/or traitors that have already obtained credentials to access a file system. Masqueraders generally include attackers that impersonate another inside user, while traitors generally include inside attackers that use their own legitimate credentials to attain illegitimate goals. In addition, some external attackers can become inside attackers when, for example, an external attacker gains internal network access. For example, external attackers can gain access to an internal network with

the use of spyware or rootkits. Such software can be easily installed on computer systems from physical or digital media (e.g., email, downloads, etc.) and can provide an attacker with administrator or “root” access on a machine along with the capability of gathering sensitive data. In particular, the attacker 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. Rootkits have the ability to conceal themselves and elude detection, especially when the rootkit is previously unknown, as is the case with zero-day attacks. An external attacker that manages to install a rootkit internally in effect becomes an insider, thereby multiplying the ability to inflict harm.

[0008] However, the masquerader is generally unlikely to know how the victim user behaves when using a file system. For example, each individual computer user generally knows his or her own file system well enough to search in a limited, targeted, and unique fashion in order to find information germane to the current task. Masqueraders, on the other hand, generally do not know the user’s file system and/or the layout of the user’s desktop. As such, masqueraders generally search more extensively and broadly in a manner that is different than the victim user being impersonated.

[0009] One approach to prevent inside attacks generally involves policy-based access control techniques that limit the scope of systems and information an insider is authorized to use, thereby limiting the damage the organization can incur when an insider goes awry. Despite these general operating system security mechanisms and the specification of security and access control policies, such as the Bell-LaPadula model and the Clark-Wilson model, the insider attacker problem is extensive. For example, in many cases, formal security policies are incomplete and implicit or they are purposely ignored in order to achieve business goals. In fact, the annual Computer Crime and Security Survey for 2007, which surveyed **494** security personnel members from corporations and government agencies within the United States, found that insider incidents were cited by about 59 percent of respondents, while only about 52 percent had encountered a conventional virus in the previous year. Other approaches have been made attempts to address these problems. However, these approaches merely perform a forensics analysis after an insider attack has occurred.

[0010] It should also be noted that, with the advent of wireless networking, the ubiquity of wireless networking exposes information to threats that are difficult to detect and defend against. Even with the latest advances aimed at securing wireless communications and the efforts put forth into protecting wireless networking, compromises still occur that allow sensitive information to be recorded, exfiltrated, and/or absconded. Secure protocols exist, such as WiFi Protected Access 2 (WPA2), that can help in preventing network compromise, but, in many cases, such protocols are not used for reasons that may include cost, complexity, and/or overhead. In fact, the 2008 RSA Wireless Security Survey reported that only 49% of corporate access points in New York, N.Y. and 48% in London, England used advanced security. Accordingly, many wireless networks remain exposed despite the existence of these secure protocols.

[0011] Moreover, one of the benefits of WiFi is the seemingly boundless, omnipresent signal. However, this broad transmission radius is also one of its greatest risks. The broadcast medium on which the suite of IEEE 802.11 protocols are

based makes them particularly difficult to secure. In general, there is little that can be done to detect passive eavesdropping on networks. This problem is exacerbated with WiFi due to the range of the signal.

[0012] In many instances, a good insider may inadvertently aid a malicious user by opening an executable file, accessing a URL, etc. that installs malicious software in a system.

[0013] There is therefore a need in the art for approaches that bait inside attackers using decoy information and measure the security of computer systems. Accordingly, it is desirable to provide methods, systems and media that overcome these and other deficiencies of the prior art.

SUMMARY

[0014] Methods, systems, and media for measuring computer security are provided. In accordance with some embodiments, methods for measuring computer security are provided, the methods comprising: making at least one of decoys and non-threatening access violations accessible to a first user using a computer programmed to do so; maintaining statistics on security violations and non-violations of the first user using a computer programmed to do so; and presenting the statistics on a display.

[0015] In accordance with some embodiments, systems for measuring computer security are provided, the systems comprising: a processor that: makes at least one of decoys and non-threatening access violations accessible to a first user; maintains statistics on security violations and non-violations of the first user; and presents the statistics on a display.

[0016] In accordance with some embodiments, non-transitory computer-readable media containing computer-executable instructions that, when executed by a processor, cause the processor to perform a method for measuring computer security, the method comprising: making at least one of decoys and non-threatening access violations accessible to a first user; maintaining statistics on security violations and non-violations of the first user; and presenting the statistics on a display.

BRIEF DESCRIPTION OF THE DRAWINGS

[0017] 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 of the disclosed subject matter.

[0018] FIG. 2 is a diagram showing an original document and a decoy document with one or more embedded beacons in accordance with some embodiments of the disclosed subject matter.

[0019] FIG. 3 is a diagram showing an example of a process for generating and inserting decoy information into an operating environment in accordance with some embodiments of the disclosed subject matter.

[0020] FIG. 4 is a diagram showing examples of actual information (e.g., network traffic) in an operating environment in accordance with some embodiments.

[0021] 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 of the disclosed subject matter.

[0022] FIG. 6 is a diagram showing an example of a process for generating decoy traffic in accordance with some embodiments of the disclosed subject matter.

[0023] FIGS. 7-8 are diagrams showing an example of an interface for managing documents containing decoy information in accordance with some embodiments of the disclosed subject matter.

[0024] FIGS. 9-11 are diagrams showing an example of an interface for generating and managing documents containing decoy information in accordance with some embodiments of the disclosed subject matter.

[0025] FIG. 12 is a diagram showing an example of a generated decoy document in the form of an eBay receipt in Microsoft Word format in accordance with some embodiments of the disclosed subject matter.

[0026] FIG. 13 is a diagram showing an example of a generated decoy document in the form of a credit card letter in Adobe PDF format in accordance with some embodiments of the disclosed subject matter.

[0027] FIG. 14 is a diagram showing an example of a generated decoy document in the form of a shopping list in accordance with some embodiments of the disclosed subject matter.

[0028] FIG. 15 is a diagram showing an example of a generated decoy document in the form of a credit card letter in Microsoft Word format in accordance with some embodiments of the disclosed subject matter.

[0029] FIG. 16 is a diagram showing an example of a generated decoy document in the form of a vacation note in accordance with some embodiments of the disclosed subject matter.

[0030] FIG. 17 is a diagram showing an example of a generated decoy document in the form of a medical billing summary in accordance with some embodiments of the disclosed subject matter.

[0031] FIG. 18 is a diagram showing an example of a generated decoy document in the form of a tax document in accordance with some embodiments of the disclosed subject matter.

[0032] FIG. 19 is a diagram showing an embedded beacon in accordance with some embodiments of the disclosed subject matter.

[0033] FIG. 20 is a diagram showing a connection opened to an external website by an embedded beacon in accordance with some embodiments of the disclosed subject matter.

[0034] FIG. 21 is a diagram showing an example of a website that collects beacon signals in accordance with some embodiments of the disclosed subject matter.

[0035] FIG. 22 is a diagram showing an example of an alert that is transmitted to a user in response to receiving signals from a beacon in accordance with some embodiments of the disclosed subject matter.

[0036] FIG. 23 is a diagram showing an example of a process for receiving signals from a beacon embedded in decoy information and removing malware in accordance with some embodiments of the disclosed subject matter.

[0037] FIG. 24 is a diagram showing an example of a process for transmitting notifications and/or recommendations in response to receiving signals from an embedded beacon in accordance with some embodiments of the disclosed subject matter.

[0038] FIG. 25 is a diagram showing an example of a process for measuring computer security in accordance with some embodiments of the disclosed subject matter.

DETAILED DESCRIPTION

[0039] In accordance with various embodiments, as described in more detail below, mechanisms for baiting inside attackers are provided. In some embodiments, systems and methods are provided that implement trap-based defensive mechanisms that can be used to confuse, deceive, and/or detect nefarious inside attackers that attempt to exfiltrate and/or use information. These traps use decoy information (sometimes referred to herein as “bait information,” “bait traffic,” “decoy media”, or “decoy documents”) to attract, deceive, and/or confuse attackers (e.g., inside attackers, external attackers, etc.) and/or malware. For example, large amounts of decoy information can be generated and inserted into the network flows and large amounts of decoy documents, or documents containing decoy information, can be generated and placed within a file system to lure potential attackers. In another example, decoy documents can be generated that are machine-generated documents containing content to entice an inside attacker into stealing bogus information. Among other things, 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.

[0040] In some embodiments, decoy information can be combined with any suitable number of monitoring or alerting approaches, either internal or external, to detect inside attackers. For example, a beacon can be embedded in a document or any other suitable decoy information. As used herein, a beacon can be any suitable code or data that assists in the differentiation of decoy information from actual information and/or assists in indicating the malfeasance of an attacker illicitly accessing the decoy information. In some embodiments, these stealthy beacons can cause a signal to be transmitted to a server indicating when and/or where the particular decoy information was opened, executed, etc.

[0041] In one example, the decoy information, such as a decoy document, can be associated and/or embedded with one or more active beacons, where the active beacons transmit signals to a remote website upon opening the document that contains the decoy information. The signals can indicate that the decoy information has been accessed, transmitted, opened, executed, and/or misused. Generally, these signals indicate the malfeasance of an insider illicitly reading decoy information. In some embodiments, the use of decoy information with the embedded active beacon can indicate that the decoy information has been exfiltrated, where the beacon signals can include information sufficient to identify and/or trace the attacker and/or malware.

[0042] In another example, the decoy information, such as a decoy document, can be associated and/or embedded with one or more passive beacons. In a more particular example, a passive beacon in the form of a watermark can be embedded in the binary format of the document file or any other suitable location of the document file format. The watermark is detected when the decoy information is loaded in memory or transmitted in the open over a network. In some embodiments, a host-based monitoring application can be configured to transmit signals or an alert when it detects the passive beacon in documents.

[0043] Alternatively, a passive beacon can be code that assists a legitimate user in differentiating decoy information from authentic information. For example, in response to opening a decoy document containing decoy information and

an embedded passive beacon, the passive beacon generates a pattern along with the decoy document. Upon placing a physical mask over the generated pattern, an indicator (e.g., a code, a sequence of letters or numbers, an image, etc.) can be displayed that allows the legitimate user to determine whether the document is a decoy document or a legitimate document.

[0044] In yet another example, the decoy information can be associated with a beacon that is both active and passive. In a more particular example, a beacon can generate a pattern, where a legitimate user can place a physical mask over the pattern to determine whether the information is decoy information or actual information, and the beacon can transmit a signal to a remote website indicating that the decoy information has been accessed.

[0045] In a further example, the content of the decoy information itself can be used to detect an insider attack. The content of the decoy information can include a bogus login (e.g., a bogus login and password for Google Mail). The bogus login to a website can be created in a decoy document and monitored by external approaches (e.g., polling a website or using a custom script that accesses mail.google.com and parses the bait account pages to gather account activity information).

[0046] As shown above, beacons can be used to detect the malfeasance of an inside attacker at any suitable time. For example, at the time of application start-up, the decoy document causes the transmission of a beacon alert to a remote server. In another example, at the time of memory load, a host-based monitoring application, such as an antivirus software application, can detect embedded beacons placed in a clandestine location of the document file format (e.g., the binary file format). In yet another example, at the time of exfiltration, a network intrusion detection system, such as Snort, can be used to detect embedded beacons during the egress or transmission of the decoy document or decoy information in network traffic. In a further example, at the time of information exploitation and/or credential misuse, monitoring of decoy logins and other credentials embedded in the document content by external systems can generate an alert that is correlated with the decoy document in which the credential was placed.

[0047] As a more particular example, in some embodiments, a deception mechanism can be provided that creates, distributes, and manages potentially large amounts of decoy information for detecting nefarious acts as well as for increasing the workload of an attacker to identify real information from bogus information. For example, the deception mechanism may create decoy documents based on documents found in the file system, based on user information (e.g., login information, password information, etc.), based on the types of documents generally used by the user of the computer (e.g., Microsoft Word documents, Adobe portable document format (PDF) files, based on the operating system (e.g., Windows, Linux, etc.), based on any other suitable approach, or any suitable combination thereof. In another suitable example, the deception mechanism may allow a user to create particular decoy documents, where the user is provided with the opportunity to select particular types of documents and particular types of decoy information. The automated creation and management of decoy information for detecting the presence and/or identity of malicious inside attackers or malicious insider activity is further described below.

[0048] As another example, in some embodiments, additionally or alternatively to creating, distributing, and/or managing decoy documents, decoy information can also be inserted into network flows. For example, the deception mechanism can analyze traffic flowing on a network, generate decoy traffic based on the analysis, and insert the decoy traffic into the network flow. The deception mechanism can also refresh the decoy traffic such that the decoy traffic remains believable and indistinguishable to inside attackers. The generation, dissemination, and management of decoy traffic of various different types throughout an operational network to create indistinguishable honeyflows are further described below.

[0049] It should be noted that, while preventive defense mechanisms generally attempt to inhibit malware from infiltrating into a network, trap-based defenses are directed towards confusing, deceiving, and detecting inside attackers within the network or external attackers and malware that have succeeded in infiltrating the network.

[0050] In some embodiments, generated decoy information can be tested to ensure that the decoy information complies with document properties that enhance the deception for different classes or types of inside attackers that vary by level of knowledge and sophistication. 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 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 entrapping a rogue system administrator or a network security staff member. In another example, decoy information can appear to contain believable, sensitive personal information and seemingly valuable information. As described further below, decoy information can be generated such that the documents are believable, variable (e.g., not repetitive, updatable such that attackers do not identify decoy information, etc.), enticing (e.g., decoy information with particular keywords or matching particular search terms), conspicuous (e.g., located in particular folders or files), detectable, differentiable from actual information, non-interfering with legitimate users, etc.

[0051] 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 password information with an embedded active 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 or malware (e.g., a zero day worm) embedded within a document. In some embodiments, the malware can be extracted to update signatures in an antivirus application or in a host-based monitoring application, search for other documents that include the same malware, etc. In yet another example, a legitimate user at a digital processing device can select and submit documents for the insertion of decoy information and beacons in order to detect and/or capture inside attackers on the

digital processing device, where the beacons allow the legitimate user to differentiate between decoy information and actual information.

[0052] Turning to FIG. 1, an example of a system 100 in which the trap-based defense can be implemented. As illustrated, 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.

[0053] 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.

[0054] 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.

[0055] 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.

[0056] Generally speaking, a user of malicious/compromised computer 110 is an inside attacker that legitimately has access to communications network 108 and/or one or more systems 102, 104, and 106, but uses his or her access to attain illegitimate goals. For example, a user of malicious/compromised computer 110 can be a traitor that uses his or her own legitimate credentials to gain access to communications network 108 and/or one or more systems 102, 104, and 106, but uses his or her access to attain illegitimate goals. In another example, a user of malicious/compromised computer 110 can be a masquerader that impersonates another inside user.

[0057] It should be noted that, in some embodiments, an external attacker can become an inside attacker when the external attacker attains internal network access. For example, using spyware or rootkits, external attackers can gain access to communications network 108. Such software

can easily be installed on computer systems from physical or digital media (e.g., email, downloads, etc.) that provides an external attacker with administrator or “root” access on a machine along with the capability of gathering sensitive data. The external attacker can also snoop or eavesdrop on one or more systems **102**, **104**, and **106** or communications network **108**, download and exfiltrate data, steal assets and information, destroy critical assets and information, and/or modify information. Rootkits have the ability to conceal themselves and elude detection, especially when the rootkit is previously unknown, as is the case with zero-day attacks. An external attacker that manages to install rootkits internally in effect becomes an insider, thereby multiplying the ability to inflict harm.

[0058] In some embodiments, the owner of malicious/compromised computer **110** may not be aware of what operations malicious/compromised computer **110** is performing 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.).

[0059] 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.

[0060] 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.

[0061] 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.

[0062] More particularly, for example, each of the one or more collaborating or client computers **102**, **104**, and **106**, malicious/compromised computer **110**, deception system **114**, and attacking computer system **116**, can be any of a general purpose device such as a computer or a special purpose device such as a client, a server, etc. Any of these general or special purpose devices can include any suitable components such as a processor (which can be a microprocessor, digital signal processor, a controller, etc.), memory, commu-

nication interfaces, display controllers, input devices, etc. For example, client computer **1010** can be implemented as a personal computer, a personal data assistant (PDA), a portable email device, a multimedia terminal, a mobile telephone, a set-top box, a television, etc.

[0063] In some embodiments, any suitable computer readable media can be used for storing instructions for performing the processes described herein, can be used as a content distribution that stores content and a payload, etc. For example, in some embodiments, computer readable media can be transitory or non-transitory. For example, non-transitory computer readable media can include media such as magnetic media (such as hard disks, floppy disks, etc.), optical media (such as compact discs, digital video discs, Blu-ray discs, etc.), semiconductor media (such as flash memory, electrically programmable read only memory (EPROM), electrically erasable programmable read only memory (EEPROM), etc.), any suitable media that is not fleeting or devoid of any semblance of permanence during transmission, and/or any suitable tangible media. As another example, transitory computer readable media can include signals on networks, in wires, conductors, optical fibers, circuits, any suitable media that is fleeting and devoid of any semblance of permanence during transmission, and/or any suitable intangible media.

[0064] 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 IEEE 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**.

[0065] Deception system **114** can be any computer, server, router, or other suitable device for modeling, generating, inserting, distributing, and/or managing 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.

[0066] For example, deception system **114** can include a decoy information broadcaster to inject decoy traffic information into a communications network. Decoy information broadcaster can be a wireless router that has the capability to support monitor mode operation (e.g., RFMON mode) and has the capability of supporting virtual interfaces (e.g., a Virtual Access Points (VAPs) feature). It should be noted that, in some embodiments, since ACK frames are recorded as part of the decoy traffic, the decoy information broadcaster can be modified to suppress IEEE 802.11 ACK frames. It should also be noted that, in some embodiments, since whole sessions are generally injected (e.g., traffic from all communicating parties including ACK frames, retransmissions, etc.), the decoy information broadcaster can be modified to ignore ACK timeouts in injected frames.

[0067] In another 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 decoy information based on that analysis, and inserts the deception information into the system 100. In yet 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.

[0068] 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.

[0069] 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.

[0070] In some embodiments, deception system 114 can be modeled based on different levels of insider sophistication and capability. For example, some inside attackers have tools available to assist in determining whether a document is a decoy document or a legitimate document, while other inside attackers are equipped with their own observations and thoughts. Deception system 114 can be designed to confuse, deceive, and/or detect low threat level inside attackers having direct observation as the tool available. The low threat level indicates that the inside attackers largely depends on what can be gleaned from a first glance. Deception system 114 can be designed to confuse, deceive, and/or detect medium threat level inside attackers that have the opportunity to perform a more thorough investigation. For example, if a decoy document contains a decoy account credential for a particular identity, the inside attacker can verify that the particular identity is real or not by querying an external system, such as a website (e.g., www.whitepages.com, www.google.com, etc.). Deception system 114 can also be designed to confuse, deceive, and/or detect high threat level inside attackers that have multiple tools available (e.g., super computers, access to informed people with organizational information). Deception system 114 can further be designed to confuse, deceive, and/or detect highly privileged threat level inside attackers that

may be aware that the system is baited with decoy information and uses tools to analyze, disable, and/or avoid decoy information.

[0071] Deception system 114 can generate decoy information and decoy documents that comply with particular properties that enhance the deception for these different classes or threat levels of inside attackers. Decoy information can be generated such that the documents are believable, enticing, conspicuous, detectable, variable, differentiable from actual or authentic information, non-interfering with legitimate users, etc.

[0072] Deception system 114 can generate decoy information that is believable. That is, decoy documents are generated such that it is difficult for an inside attacker to discern whether the decoy document is an authentic document from a legitimate source or if the inside attacker is indeed looking at a decoy document. 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.

[0073] 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.

[0074] It should be noted that, in some embodiments, deception system 114 can search through files on a computer (e.g., one or more of collaborating systems 102, 104, and 106), receive templates, files, or any other suitable input from a legitimate user (e.g., an administrator user) of a computer, monitor traffic on communications network 108, or use any other suitable approach to create believable decoy information. For example, deception system 114 can determine which files are generally accessed by a particular user (e.g., top ten, last twenty, etc.) and generate decoy information similar to those files. In another example, deception system 114 can perform a search and determine various usernames, passwords, credit card information, and/or any other sensitive information that may be stored on one or more of collaborating system 102, 104, and 106. Deception system 114 can then create receipts, tax documents, and other form-based documents with decoy credentials, realistic names, addresses, and logins. In some embodiments, deception system 114 can monitor the file system and generate decoy documents with file names similar to the files accessed on the file system (e.g., a tax document with the file name "2009 Tax Form-1099-1") or with file types similar to the files accessed on the file system (e.g., PDF file, DOC file, URL link, HTML file, JPG file, etc.).

[0075] It should also be noted that, in accordance with some embodiments, decoy information can include any suitable data that is used to entrap 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 pass codes. 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).

[0076] It should be noted that, in some embodiments, deception system 114 does not request, gather, or store personally identifiable information about the user (e.g., a user of one of collaborating systems 102, 104, and 106). For example, deception system 114 does not gather and store actual password information associated with a legitimate user.

[0077] In some embodiments, deception system 114 can determine whether decoy information, such as a decoy document, complies with a believability property. Deception system 114 can test generated decoy documents to measure the believability of the document. For example, deception system 114 can perform a decoy Turing test, where two documents are selected—one document is a decoy document and the other document is randomly selected from a collection of authentic documents (e.g., an authentic document on a computer, one of multiple authentic documents selected by a user of the computer, etc.). The two documents can be presented to a volunteer or any other suitable user and the volunteer can be tasked to determine which of the two documents is authentic. In some embodiments, in response to testing the believability of a decoy document and receiving a particular response rate, deception system 114 can consider the decoy document to comply with the believability property. For example, deception system 114 can determine whether a particular decoy document is selected as an authentic document at least 50% of the time, which would be the probability if the volunteer user were to select documents at random. In another example, deception system 114 can allow a user, such as an administrator user, to select a particular response rate for the particular type of decoy document. If the decoy document is tested for compliance with the believability property and receives an outcome less than the predefined response rate, deception system 114 can discard the decoy document and not insert the decoy document in the file system or the communications network.

[0078] In another example, a decoy Turing test can be conducted on generated decoy traffic, which relies upon users to distinguish between authentic and machine-generated decoy network traffic. An inability to reliably discern one traffic source from the other attests to decoy believability. For the decoy Turing test, traffic from multiple hosts on a private network can be recorded. The test users can be instructed to access the private network and engage one another in email conversations, use the Internet, conduct file transfer protocol (FTP) transactions, etc. The recorded traffic can include, for example, HTTP traffic, Gmail account activity, POP, and SMTP traffic. Deception system 114 can then scrub non-TCP traffic to reduce the volume of data and the resulting trace can be passed to the decoy traffic generation process described below. Honeyflows can be loaded with decoy credentials, given their own MAC and IP addresses, and then interwoven

with the authentic flows to create a file containing all of the network trace data. Each user can then be asked to determine whether traffic is authentic traffic or decoy traffic.

[0079] Alternatively, deception system 114 can decrease the response rate for a decoy document as an inside attacker generally has to open the decoy document to determine whether the document is an authentic document or not. The inside attackers can be detected or trapped in response to opening, transmitting, and/or executing the decoy document prior to determining the believability of the document.

[0080] Deception system 114 can also generate decoy information that is enticing. That is, a decoy document can be generated such that it attracts inside attackers to access, transmit, open, execute, and/or misuse the decoy document. For example, deception system 114 can generate decoy documents containing information with monetary value, such as passwords or credit card numbers. In another example, to entice a sophisticated and knowledgeable inside 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 entrapping a system administrator or a network security staff member. In yet another example, deception system 114 can monitor the file system and generate decoy documents with file names containing particular keywords (e.g., stolen, credit card, private data, Gmail account information, tax, receipt, statement, record, medical, financial, password, etc.).

[0081] In some embodiments, in addition to modifying the content of the actual information, additional content can be inserted into the decoy information to entice 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.

[0082] In some embodiments, deception system 114 can create categories of interest for inside attackers and generate decoy documents containing decoy information assigned to one or more of the categories of interest. Categories of interest can include, for example, financial, medical record, shopping list, credit card, budget, personal, bank statement, vacation note, or any other suitable category. For an inside attacker interested in financial information, deception system 114 can create enticing decoy documents that mention or describe information that provides access to money. In another example, the user of a computer can select one or more categories of interest which the user desires to protect from inside attackers, such as login information, financial information, and/or personal photographs. In response, deception system 114 can generate, for example, a “password” note in Microsoft Outlook that contains decoy usernames and passwords for various websites, a W-2 tax document in Adobe PDF format that contains decoy tax and personal information, and a series of images obtained from Google Images with enticing filenames. In yet another example, deception system 114 can determine frequently occurring search terms associated with particular categories of interest (e.g., the terms “account” and “password” for the login information category).

[0083] In some embodiments, deception system 114 can create enticing documents for insertion into a file system. For example, deception system 114 can monitor the file system and generate decoy documents with file names similar to the files accessed on the file system (e.g., a tax document with the file name “2009 Tax Form-1099-1”).

[0084] In some embodiments, deception system 114 can determine whether decoy information, such as a decoy document, complies with the enticing property. Deception system 114 can test generated decoy documents to determine whether the document is enticing to an inside attacker. For example, deception system 114 can perform content searches on a file system or network that contains decoy documents and count the number of times decoy documents appear in the top ten list of documents. In response to testing how enticing a decoy document is and receiving a particular count, deception system 114 can consider the decoy document to comply with the enticing property. For example, deception system 114 can determine whether a particular decoy document appears as one of the first ten search results. In another example, deception system 114 can allow a user, such as an administrator user, to select a particular count threshold for the particular type of decoy document or category of interest. If the decoy document is tested for compliance with the enticing property and receives an outcome less than the particular count threshold, deception system 114 can discard the decoy document and not insert the decoy document in the file system or the communications network.

[0085] It should be noted that, as enticement can depend upon the attacker's intent or preference, enticing information can be defined in terms of the likelihood of an adversary's preference and enticing decoy information can be information of those decoys that are chosen with the same likelihood.

[0086] It should be also noted that, in some embodiments, these enticing decoy documents 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 can be 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.

[0087] Deception system 114 can also generate decoy information that is conspicuous. That is, a decoy document can be generated such that it is easily found or observed on a file system or a communications network. For example, deception system 114 can place decoy documents on the desktop of a computer. In another example, deception system 114 can place decoy a document such that the document is viewable after a targeted search action.

[0088] In some embodiments, deception system 114 can place the decoy document in a particular location selected from a list of locations associated with the category of decoy document. For example, a decoy tax document can be placed in a "Tax" folder or in the "My Documents" folder. Alternatively, deception system 114 can insert the decoy document in a randomly selected location in the file system.

[0089] In some embodiments, deception system 114 can determine whether decoy information, such as a decoy document, complies with the conspicuous property. Deception system 114 can test generated decoy documents to determine whether the document is easily visible to an inside attacker. For example, deception system 114 can perform a query and count the number of search actions needed, on average, for the decoy document to appear. The query can be a search for a location (e.g., a search for a directory named "Tax" in which

the decoy document appears) and/or a content query (e.g., using Google Desktop Search for documents containing the word "Tax").

[0090] Based on the count, deception system 114 can determine whether the decoy document is to be placed at a particular location (e.g., a folder on the desktop named "Tax") or stored anywhere in the file system (e.g., not in a specific folder). For example, deception system 114 can determine that the decoy document can be stored anywhere in the file system if a content-based search locates the decoy document in a single step.

[0091] It should be noted that, for the document space M, deception system 114 can create a variable V as the set of documents defined by the minimum number of user actions required to enable their view. A user action can be any suitable command or function that displays files and documents (e.g., ls, dir, search, etc.). A subscript can be used to denote the number of user actions required to view some set of documents. For example, documents that are in view at logon or on the desktop, which require no user actions, are labeled V_0 . In another example, documents requiring one user action are labeled V_1 . A view V_i of a set of documents can be defined as a function of a number of user actions applied to a prior view, V_{i-1} , or:

$$V_i = \text{Action}(V_{i-1}), \text{ where } V_j \neq V_i, j < i$$

[0092] In some embodiments, in response to testing the conspicuous property of a decoy document and receiving a particular probability, deception system 114 can consider a decoy document to comply with the conspicuous property. For example, if a decoy document is placed on the desktop, deception system 114 can consider such a document in full view as highly conspicuous. In another example, deception system 114 can allow a user, such as an administrator user, to select a particular probability (e.g., $P=75\%$) or view (e.g., only V_0 and V_1 documents) for the particular type of decoy document.

[0093] Deception system 114 can also generate decoy information that is detectable. Deception system 114 can combine decoy information with any suitable number of monitoring or alerting approaches, either internal or external, to detect inside attackers.

[0094] In one suitable embodiment, deception system 114 can associate and/or embed a decoy document with one or more beacons. As described above, a beacon can be any suitable code or data that assists in the differentiation of decoy information from actual information and/or assists in indicating the malfeasance of an attacker illicitly accessing the decoy information. For example, at the time the application starts up and opens a decoy document, a beacon in a decoy document can transmit an alert to a remote server. The beacon can transmit a signal that includes information on the inside attacker to a remote website upon accessing the document that contains the decoy information. The signal can also indicate that the decoy information has been transmitted, opened, executed, and/or misused. In another example, the embedded beacon can indicate that the decoy information has been exfiltrated, where the beacon signals can include information sufficient to identify and/or trace the attacker and/or malware.

[0095] In another suitable embodiment, deception system 114 can implement one or more beacons in connection with a host sensor or a host-based monitoring application, such as 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. In another example, the host-based monitoring application can detect embedded passive beacons or tokens placed in a clandestine location of the document file format. In particular, a passive beacon, such as a watermark, can be embedded in the binary format of the document file to detect when the decoy information is loaded into memory. By embedding specific beacon signatures in the decoy documents, the host-based monitoring application can detect and/or receive beacon signals each time the decoy documents are accessed, opened, etc. Information about the purloined document can be uploaded to the host-based monitoring application. In yet another example, deception system **114** can implement a beacon that is both active and passive. That is, in one example, a passive portion of a beacon can generate a pattern, where a legitimate user can place a physical mask over the pattern to determine whether the information is decoy information or actual information, and an active portion of the beacon can transmit a signal to a remote website indicating that the decoy information has been accessed.

[0096] For example, 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 discernable 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, to ensure that the decoy document is enticing without including personally identifying information, 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.

[0097] In yet another suitable embodiment, deception system **114** can implement one or more beacons in connection with a network intrusion detection system. A network intrusion detection system, such as Snort, can be used to detect these embedded beacons or tokens during the egress or exfiltration of the decoy document in network traffic.

[0098] In some embodiments, a decoy document itself can be used to detect inside attackers at the time of information exploitation and/or credential misuse. For example, the content of decoy information can include a decoy login (e.g., a decoy login and password for Google Mail) and/or other credentials embedded in document content. The bogus login to a website can be created in a decoy document and can be monitored by external approaches (e.g., using a custom script that accesses mail.google.com and parses the bait account pages to gather account activity information). Monitoring the use of decoy information by external systems (e.g., a local IT system, at Gmail, at an external bank, etc.) can be used to generate an alert that is correlated with a decoy document in which a credential was placed. For example, an alert can be generated in response to an attacker logging in using a decoy login and/or performing any other suitable action (e.g., changing the password on a bogus Gmail account).

[0099] For example, if deception system **114** creates unique decoy usernames for each computer in system **100**, the use of a unique decoy username can assist deception system **114** to determine which computer has been compromised, the identity of the inside attacker, etc. Deception system **114** can discover the identity and/or the location of attacking computer systems (e.g., attacking computer system **116**). Decep-

tion 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 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.

[0100] It should be noted that, 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.

[0101] In some embodiments, deception system **114** can be used to educate and/or train users to reduce user errors or user mistakes. For example, an organization can routinely or at random present to its employee users a stream of decoy information to test whether one of the employee users accesses one or more pieces of decoy information, thereby violating the organization's policy. In response to accessing decoy information, any suitable action can be performed, such as contacting the IT department, sending an email notification to the employee user that accessed the decoy information, directing the employee user for additional training, etc. In another example, the transmission of emails with decoy URLs or emails with decoy documents that, if opened, sound an alarm, or embedded decoy data in databases that, upon extraction, reveal a policy violation, can be used to educate users, refresh decoy information, and refresh or restate organizational policies, thereby reducing accidental insider threats.

[0102] Deception system **114** can also generate decoy information that is variable. That is, decoy documents can be generated such that they are not easily identifiable due to some common invariant information shared between decoy documents. For example, decoy documents that are varied are those in which a single search or test function does not easily distinguish actual documents from decoy documents. In particular, if the same sentence appears in 100 decoy documents, decoy documents with such repetitive information may not be considered to comply with the variability property.

[0103] Deception system **114** can also generate decoy information that does not interfere with regular operations of a legitimate user and is differentiable. That is, deception system **114** can generate decoy documents that, for an inside attacker, are indistinguishable from actual documents, but also do not ensnare the legitimate user. To comply with the non-interfering property, deception system **114** can create

decoy documents so that a legitimate user does not accidentally misuse the bogus information contained within the decoy document.

[0104] In some embodiments, deception system 114 can determine whether decoy information, such as a decoy document, complies with the non-interfering property. Deception system 114 can determine the number of times a legitimate user accidentally accesses, executes, transmits, and/or misuses a decoy document. For example, deception system 114 can include an alert component that transmits an email to the legitimate user each time a decoy document is accessed, executed, transmitted, etc. In response to receiving the alert (e.g., an email message), the user can be prompted to indicate whether the alert is a false alarm such that the legitimate user accidentally accessed, executed, transmitted, and/or misused the decoy document. Deception system 114 can then monitor the number of times a false alarm is created and, based on the monitoring, determine whether a particular decoy document complies with the non-interfering property. For example, in response to receiving more than three false alarms for a particular decoy document, deception system 114 can perform a suitable action—e.g., rename the decoy document, remove the decoy document from the file system, request that the legitimate user provide suggestions to modify the decoy document (e.g., to not ensnare the legitimate user again), etc.

[0105] In some embodiments, a computational object (e.g., a function) can be embedded in each document on a file system and each decoy document inserted in the file system. When a document is executed (e.g., in response to a mouse click), deception system 114 can generate and display a pattern on a display monitor in a bounded box. Alternatively, the pattern generated by the embedded computational object can be rendered or produced by an application associated with the document. For each decoy document and each actual document, the display can vary in such a way that an observer can distinguish between real and decoy documents using a physical mask, such as a uniquely patterned transparent screen. When the physical mask is overlaid on the displayed window containing the generated pattern, a derived word, picture, icon, or any other suitable indicia can be revealed that allows the legitimate user to discriminate between real and bogus decoys. Accordingly, to discern decoy documents from non-decoy documents, an attacker has to steal the user's document files stored on a hard drive or file system and the physical mask.

[0106] It should be noted that the execution of the embedded computational objects in each document produces randomly appearing patterns in windows that are not differentiable without the user's physical mask. For example, if one hundred decoy documents for each real document are stored on the file system, the attacker's task is daunting as the attacker needs to test embedded information in the documents to decide what is real and what is bogus, thereby thwarting these nefarious acts.

[0107] It should also be noted that the physical mask can be associated with a code unique to the particular user or to the particular application. For example, the physical mask can have a pattern imprinted on it that is keyed by a unique code (e.g., derived or linked to the serial number of the document application used to create the real documents). In response to placing that physical mask over the display window, information that indicates the document is real can be revealed.

[0108] Alternatively, in some embodiments, a legitimate user can differentiate between legitimate documents and decoy documents using an authentication or integrity code and a keycode. For example, each document, whether a legitimate document or a decoy document, can be associated with an authentication code or integrity code. The keycode or another suitable secret key assists the legitimate user in determining whether a document is legitimate or a decoy document. In some embodiments, one or more codes can be placed in a document (e.g., hidden in the document, conspicuous, etc.).

[0109] For example, a function can be defined that generates a secret regular language described by a regular expression, R . R can be defined by some alphabet over 36 symbols (26 letters, 10 numbers). In addition, R can be randomly generated and can be used as a pattern to decide whether a token is a member of the language or not. Deception system 114 can randomly generate strings from $L(R)$ each time a decoy document is created. Each random string is embedded in the decoy document. In another example, the interface that generates the decoy documents and the document generation application (e.g., an Adobe PDF generator) can be coordinated to embed documents generated by the user's machine tokens from the language $\sim L(R)$ and randomly generate tokens from this language to be embedded in each legitimate document. Accordingly, to determine whether a document is a legitimate document or a decoy document, the embedded token can be tested to determine whether it is a member of $L(R)$ or its complement $\sim L(R)$.

[0110] In another example, hash functions can be used in connection with the decoy and legitimate documents. For example, watermarks can appear as a hash of pseudo-randomly selected parts of the document and a secret key known only to the legitimate user. For the decoy documents, an invalid hash (e.g., that does not verify) is also computed. With a cryptographically strong hash function and with a secret key known only to the legitimate user, there is little for the inside attacker to learn.

[0111] Although some embodiments are described as decoding a generated pattern with a physical mask, it should be noted that this is merely illustrative. To allow a legitimate user to differentiate decoy information from actual information, the legitimate user can use a scanning or decoding device (e.g., a camera phone) or any other suitable device that is associated with the legitimate user. For example, the legitimate user can register a particular cellular telephone with deception system 114. In response to the legitimate user accessing a decoy document, the passive beacon associated with the decoy document can generate a pattern, such as a unique three-dimensional bar code or a machine-readable number that identifies the particular document. Upon using an application on the device to capture an image of the pattern and transmit at least a portion of the image (or corresponding data) to a server (e.g., deception system 114), the legitimate user can be provided with an indication as to whether the document is a decoy document or an actual document (e.g., a graphic displayed on the camera phone, a text message, etc.). Accordingly, similar to the physical mask, to discern decoy documents from non-decoy documents, an attacker has to steal the user's document files stored on a hard drive or file system and the decoding device associated with the user.

[0112] Accordingly, decoy information that complies with one or more of the above-mentioned properties can be used to confuse and/or slow down an inside attacker or an attacker

using attacking computer system **116**. For example, an inside attacker or an attacker at 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.

[0113] As described above, trap-based defenses using decoy information can be provided to users of collaborating systems **102**, **104**, and/or **106**, malicious/compromised computer **110**, and/or communications network **108** of FIG. **1**. FIG. **3** illustrates an example **300** of a process for providing trap-based defenses in accordance with some embodiments of the disclosed subject matter. 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 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. As shown, IP addresses, source and destination MAC IP addresses, identifying attributes, credentials, usernames, passwords, and other suitable information 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.

[0114] 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 entrap attackers and/or malware. Decoy information can include user behavior at the level of 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.

[0115] Illustrative examples of decoy traffic information and honeyflows are shown in FIG. **5**. 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 that decoy account usernames, decoy account passwords, decoy media access control (MAC) addresses, modified IP addresses, modified protocol commands, etc. have been generated and inserted into the communications network. The decoy information can be used to entice attackers and/or malware seeking to access and/or misuse the decoy information.

[0116] As a more particular example, an example **600** of a process for generating decoy traffic is shown in FIG. **6** in accordance with some embodiments of the disclosed subject matter. As shown, monitored and/or recorded trace data can be inputted into deception system **114** at **610**. For example,

one or more templates, each containing anonymous trace data, can be provided to deception system **114**. In another example, a complete network trace containing authentic network traffic can be provided to deception system **114**.

[0117] It should be noted that, regarding the recordation of network traffic, deception system **114** can receive either anonymous trace data or authentic network traffic. For example, within a university environment or any other suitable environment in which there may be concerns (e.g., ethical and/or legal) regarding the recordation of network traffic, one or more templates containing anonymous trace data can be created. These can be protocol-specific templates that contain TCP session samples for protocols used by the decoys. Alternatively, in environments having privacy concerns, deception system **114** can record a specific sample of information, events, and traffic (e.g., information that does not include personally identifying information).

[0118] In environments in which there are no concerns regarding the recordation of network traffic (e.g., enterprise environments), live network traces can be provided to deception system **114**. In these situations, 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, for example. 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.

[0119] At **620**, in response to receiving the inputted network data, the protocol type of the trace data can be determined based at least in part on the content of the trace data. Deception system **114** can, using one or more pre-defined rules, analyze the inputted trace data to determine protocol types based on the content of application layer headers. That is, deception system **114** can examine header identifiers within the trace data, where the header identifiers are specific for a given protocol. For example, application layer headers, such as "AUTH PLAIN", "EHLO", "MAIL FROM:", "RCPT TO:", "From:", "Reply-To:", "Date:", "Message-Id:", "250", "220", and "221", can be used to identify that the particular portion of trace data uses the Simple Mail Transfer Protocol (SMTP).

[0120] At **630**, one or more candidate flows for each protocol type can be generated. For example, if the inputted network data matches criteria of pre-defined rule sets, deception system **114** can separate the inputted network data and create a set of candidate flows including authentication cookies, HTTP traffic, documents, and/or SMTP, POP, IMAP, or FTP credentials. At **640**, one or more rules can be applied to modify the candidate flows with decoy information. For example, deception system **114** can support rules for adding decoy information or bait into protocol headers (e.g., IP addresses, SMTP passwords, etc.) and protocol payloads (e.g., the body of emails, web page content, etc.). Different types of decoy traffic can be created, such as Gmail authentication cookies, URLs, passwords for unencrypted protocols as SMTP, POP, and IMP, and beacons documents as email attachments. The generation of decoy documents is described in further detail below.

[0121] 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.

[0122] In some embodiments, deception system 114 and/or the decoy information broadcaster can refresh the decoy traffic such that the decoy traffic remains believable and indistinguishable to inside attackers. For example, one type of decoy traffic is authentication cookies, which are generally valid for a finite amount of time. In response, such decoy traffic can be refreshed after a predetermined amount of time has elapsed (e.g., every minute, every day, etc.). It should be noted that, if the same decoy traffic were continuously replayed within the communications network, an inside attacker would be able to distinguish the decoy traffic from authentic traffic based on the retransmissions of protocol header portions (e.g., TCP sequence numbers, IP time to live (TTL), TCP/UDP source port numbers, IP identifiers (ID), etc.). In one example, new honeyflows containing new and/or refreshed decoy traffic information are generated at deception system 114 and transmitted to one or more decoy information broadcasters for insertion into their associated communications network. Alternatively, in another example, each decoy information broadcaster generates new honeyflows containing new and/or refreshed decoy traffic information and those honeyflows are inserted into its associated communications network.

[0123] In addition to inserting decoy information, these honeyflows or traffic containing decoy information can be modified to create variability and randomness. Deception system 114 can perform a rule-driven replacement of MAC addresses and IP addresses to those from predefined set (e.g., a list of decoy MAC addresses, a list of decoy IP addresses, etc.) in some embodiments. Deception system 114 can also use natural language programming heuristics to ensure that content matches throughout the decoy traffic or decoy document. For example, deception system 114 can ensure that content, such as names, addresses, and dates, match those of the decoy identities.

[0124] In some embodiments, deception system 114 can support the parameterization of temporal features of the communications network (e.g., total flow time, inter-packet time, etc.). That is, deception system 114 can extract network statistics from the network data (e.g., the inputted trace data) or obtain network statistics using any suitable application. Using these network statistics, deception system 114 can modify the decoy traffic such that it appears statistically similar to normal traffic.

[0125] In some embodiments, deception system 114 can obtain additional information relating to collaborating systems 102, 104, and/or 106, malicious/compromised computer 110, and/or communications network 108 of FIG. 1 on which deception system 114 is generating decoy traffic. For example, deception system 114 can determine the operating system of the computer (e.g., using OS fingerprint models) to

generate decoy information that is accurately modeled for a given host operating system. To generate decoy traffic that appears to emanate from a Linux host, email traffic can be generated that appears to have come from the Evolution email client, as opposed to Microsoft Outlook that is generally used on devices where Microsoft Windows is the operating system.

[0126] 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. It should be noted that use of the historical information can be localized to specific collaborating systems or specific network segments to inhibit the exposure of sensitive information. For example, recorded historical information in one subnet may not be used in another subnet to avoid exposing sensitive information that would otherwise remain hidden from malware located in one of the subnets.

[0127] In some embodiments, snapshots of a 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.

[0128] Upon generating decoy traffic, deception system 114 can inject the decoy traffic into a communications network. As described above, deception system 114 can include a decoy information broadcaster to inject decoy traffic information into a communications network. Decoy information broadcaster can be a wireless router that has the capability to support monitor mode operation (e.g., RFMON mode) and has the capability of supporting virtual interfaces (e.g., a Virtual Access Points (VAPs) feature). It should be noted that, in some embodiments, since ACK frames are recorded as part of the decoy traffic, the decoy information broadcaster can be configured to suppress 802.11 ACK frames. It should also be noted that, in some embodiments, since whole sessions are generally injected (e.g., traffic from all communicating parties including ACK frames, retransmissions, etc.), the decoy information broadcaster can also be configured to ignore ACK timeouts in injected frames.

[0129] In response to configuring the decoy information broadcaster, a virtual access point can be created and the created virtual access point can be set to monitor mode. The generated decoy traffic can be transferred to the decoy information broadcaster, where tcpplay or any other suitable tool can be used to playback or disperse the decoy traffic inside the communication, network associated with the decoy information broadcaster.

[0130] As mentioned above, deception system 114 and/or the decoy information broadcaster can refresh the decoy traffic such that the decoy traffic remains believable and indistinguishable to inside attackers. For example, one type of decoy traffic is authentication cookies, which are generally valid for a finite amount of time. In response, decoy traffic can be refreshed after a predetermined amount of time has elapsed (e.g., every minute, every day, etc.). It should be noted that, if the same decoy traffic were continuously replayed within the communications network, an inside attacker may be able to distinguish the decoy traffic from authentic traffic based on the retransmissions of protocol header portions (e.g., TCP sequence numbers, IP time to live (TTL), TCP/

UDP source port numbers, IP identifiers (ID), etc.). In one example, new honeyflows containing new and/or refreshed decoy traffic information are generated at deception system 114 and transmitted to one or more decoy information broadcasters for insertion into their associated communications network. Alternatively, in another example, each decoy information broadcaster generates new honeyflows containing new and/or refreshed decoy traffic information and those honeyflows are inserted into its associated communications network. The determination between using deception system 114 or the decoy information broadcaster to generate and/or refresh the decoy traffic may be based on, for example, the processing power of the decoy information broadcaster, the delay between the time that deception system 114 decides to generate and transmit decoy traffic and the time that the actual injection into the communications network takes place, etc.

[0131] In some embodiments, deception system 114 can support the parameterization of temporal features of the communications network (e.g., total flow time, inter-packet time, etc.). That is, deception system 114 can extract network statistics from the inputted network data or obtain network statistics using any suitable application. Using these network statistics, deception system 114 can modify the decoy traffic such that it appears statistically similar to normal traffic.

[0132] In some embodiments, deception system 114 can embed beacons along with the decoy traffic or portions of the decoy traffic. For example, passive beacons can be used that allow a monitoring application to detect the transmission of decoy traffic over the network. In another example, decoy documents that are generated as a portion of the decoy traffic can be embedded with active beacons, where the active beacons transmit a signal to a remote website or the monitoring application in response to an attacker accessing the decoy document from the decoy traffic.

[0133] As another example, in some embodiments, additionally or alternatively to generating, inserting, and/or managing honeyflows and decoy information in network flows, a deception mechanism can be provided that creates, distributes, and manages decoy information for detecting nefarious acts as well as to increase the workload of an attacker to identify real information from bogus information. For example, the deception mechanism may create decoy documents based on documents found in the file system, based on user information (e.g., login information, password information, etc.), based on the types of documents generally used by the user of the computer (e.g., Microsoft Word documents, Adobe portable document format (PDF) files, based on the operating system (e.g., Windows, Linux, etc.), based on any other suitable approach, or any suitable combination thereof. In another suitable example, the deception mechanism may allow a user to create particular decoy documents, where the user is provided with the opportunity to select particular types of documents and particular types of decoy information.

[0134] FIGS. 7-18 show a deception mechanism for creating, distributing, and/or managing decoy documents in accordance with some embodiments of the disclosed subject matter. In some embodiments, decoy information and, more particularly, decoy documents 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. In another example, a user of a computer can provide documents, whether exemplary documents or templates, for the creation of decoy documents. Accordingly, using an interface, a user (e.g., government intelligence officer, an information technology professional, etc.) can create tailored decoy information, such as a top secret jet fighter design document or a document that includes a list of intelligence agents.

[0135] Turning to FIG. 7, a website or any other suitable interface can be provided to a user for generating, obtaining (e.g., downloading), and managing decoy documents in accordance with some embodiments. As shown in FIG. 7, the website requests that the user register with a legitimate email address (e.g., user@email.com). In response to registering with the website and entering in the legitimate email address along with a password, the website provides the user with the opportunity to create and/or download decoy documents, load user-selected documents or customized documents for the insertion of one or more beacons, and/or view alerts from beacons embedded in generated decoy documents, as shown in FIG. 8.

[0136] In response to the user selecting to generate a decoy document (e.g., pre-existing decoy documents that have embedded beacons, using decoy document templates), deception system 114 can provide an interface that allows the user to generate customized decoy documents for insertion into the file system. An exemplary interface is shown in FIGS. 9-11. As shown, display 900 provides the user with fields 910 and 920 for generating decoy documents. Field 910 allows 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, an HTML file, etc.) (See, e.g., FIG. 10). Field 920 allows the user to select a particular 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, a vacation note, a credit card statement, a medical record, etc.) (See, e.g., FIG. 11).

[0137] In some embodiments, the exemplary interface shown in FIGS. 9-11 can allow the user to input suggested content for insertion in the decoy documents. For example, the user can input a particular user name and/or company name for use in the decoy document. In another example, the user can input a particular file name or portion of a file name for naming the decoy document. Alternatively, the user can indicate that a random user and/or company for inclusion in the decoy document can be selected.

[0138] In some embodiments, the exemplary interface shown in FIGS. 9-11 can access publicly available documents that can be obtained using search engines, such as www.google.com and www.yahoo.com, to generate decoy information. For example, the user can select that the interface of deception system 114 obtain one or more PDF-fillable tax forms from the www.irs.gov website. In another example, the user can select that the interface of deception system 114 search one or more computers for exemplary documents and/or information for conversion into decoy documents.

[0139] In response to the user selecting one or more options (e.g., type, theme, etc.) and selecting, for example, a generate button 930 (or any other suitable user interface), the interface generates a decoy document and provides the decoy document to the user. For example, the above-mentioned decoy document properties assist the interface to design decoy

document templates and the decoy document templates are used to generate decoy documents. The content of each decoy document includes one or more types of bait or decoy information, such as online banking logins provided by a collaborating financial institution, login accounts for online servers, and web-based email accounts. As shown in FIGS. 9-11, the generated decoy documents are provided in a list 940, where the user is provided with the opportunity to download one or more decoy documents. Upon obtaining the generated decoy documents, the user can insert the decoy documents into the user's local machine, another user's local machine, place the document on a networked drive, etc.

[0140] Illustrative examples of generated decoy documents are shown in FIGS. 12-18. As shown, decoy documents can include an eBay receipt in Word format (FIG. 12), a credit card letter in PDF format (FIG. 13) and in Word format (FIG. 15), a shopping list (FIG. 14), a vacation note in Word format (FIG. 16), a medical billing summary (FIG. 17), and an Internal Revenue Service Form 1040 tax document (FIG. 18). As shown in FIGS. 12-18, the interface has generated multiple decoy documents that include decoy customer information (e.g., names, addresses, credit card numbers, tracking numbers, credit card expiration dates, salary numbers, tax information, social security numbers, payment amounts, email addresses, etc.).

[0141] Referring back to FIG. 8, the exemplary interface provides a user with the opportunity to load user-selected or customized documents. For example, the user can select forms (e.g., blank PDF fillable forms), templates, actual documents, and/or any other suitable document for use in generating decoy documents.

[0142] It should be noted that, although some of the embodiments described herein generate decoy documents based on user-selected document types, user-selected theme, and/or user-identified documents on a file system, these are illustrative. For example, in some embodiments, deception system 114 can generate decoy documents based on a search of the user computer. For example, deception system 114 may search and/or monitor a computer to determine documents found on the system, top ten documents accessed by a particular user, etc.

[0143] It should also be noted that, in some embodiments, the interface of deception system 114 can monitor the amount of time that a particular decoy documents remains on a file system and, after a particular amount of time has elapsed, refresh the decoy documents and/or send a reminder to the user to generate new decoy documents. For example, in response to a medical record decoy document remaining on a particular file system for over 90 days, deception system 114 can generate a reminder (e.g., a pop-up message, an email message, etc.) that requests that the user allow the deception system 114 to refresh the decoy document or requests that the user remove the particular decoy document and generate a new decoy document.

[0144] In some embodiments, alternatively or additionally to allowing the user to download the decoy documents into the file system, the interface can instruct the user to place the decoy document in a particular folder. For example, the interface can recommend 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.). Alternatively, the interface can insert one or more decoy documents into particular locations on the file system.

[0145] In some embodiments, the interface can provide a user with information that assists the user to more effectively deploy the decoy documents. The interface can prompt the user to input information suggestive of where the deception system or any other suitable application can place the decoy documents to better attract potential attackers. For example, the user can indicate that the decoy information or decoy document be placed in the "My Documents" folder on collaborating system. In another example, the interface can instruct the user to create a folder for the insertion of decoy document, such as a "My Finances" folder or a "Top Secret" folder.

[0146] In some embodiments, the interface 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 interface to analyze the user's computer, the website can record particular documents from the user's computer and generate customized decoy documents. In some embodiments, in response to the user allowing the interface to analyze the user's computer, the interface can provide a list of recommended folders to place decoy media.

[0147] In some embodiments, each collaborative system (e.g., collaborating 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 evenly 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 stored on each collaborative system (e.g., 10% of the actual information).

[0148] In some embodiments, the interface 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 interface can monitor for attempts by users to input a decoy credit card number. In response to receiving a decoy credit card number, the interface 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 interface 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.

[0149] In some embodiments, decoy information can be combined with any suitable number of monitoring or alerting approaches, either internal or external, to detect inside attackers. Referring back to FIG. 3, in some embodiments, one or more beacons (e.g., active beacons, passive beacons, watermarks, a code that generates a pattern, etc.) can be associated with and/or embedded into the generated decoy information at 306. Generally speaking, a beacon can be any suitable code (executable or non-executable) or data 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 and/or that assists in the differentiation of decoy information from actual information. Next, at 308, the decoy information along with the embedded beacons can be inserted into the operating environment.

[0150] 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 transmits information about the attacker to a website. In a more particular example, in response to an attacker opening a decoy Microsoft Word document entitled "2009 Tax 1099," a beacon in the form of a macro is automatically triggered and that beacon transmits a signal to a remote website. More particularly, a local browser application can be invoked from within a Word macro and information, such as local machine directories, user's credentials, and the machine's IP address can be encoded and passed through a firewall by the local browser agent. The website can then, for example, transmit an email notification to a legitimate user associated with the opened decoy document. In yet another example, the Adobe Acrobat application include a Javascript interpreter that can issue a data request upon the opening of the decoy document through the use of Javascript code. The beacon contains the token to identify the document so that deception system 114 can track individual documents as they are read across different systems.

[0151] In some embodiments, the beacon is a passive beacon, such as an embedded code or a watermark code that is detected upon attempted use. For example, the beacon is an embedded mark or a code hidden in the decoy media or document that is scanned during the egress or transmission of the decoy media or document in network traffic. In another example, the beacon is an embedded mark or a 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.

[0152] In some embodiments, the beacon is both an active beacon and a passive beacon. For example, a passive portion of the beacon can generate a pattern, where a legitimate user can place a physical mask over the pattern to determine whether the information is decoy information or actual information, and the active portion of the beacon can transmit a signal to a remote website indicating that the decoy information has been accessed.

[0153] The signals emitted from a beacon (or from an application that executes the decoy information containing the beacon) can 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, where the beacon signals can include information sufficient to identify and/or trace the attacker and/or malware. In yet another suitable example, the content of the decoy information itself can be used to detect an insider

attack. The content of the decoy information can include a bogus login (e.g., a bogus user id and password for Google Mail). The bogus login to a website can be created in a decoy document and monitored by external approaches (e.g., using a custom script that accesses mail.google.com and parses the bait account pages to gather account activity information).

[0154] In another suitable embodiment, deception system 114 can implement one or more beacons in connection with a host sensor or a host-based monitoring application, such as 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. In another example, the host-based monitoring application can detect embedded beacons or tokens placed in a clandestine location of the document file format. In particular, a watermark can be embedded in the binary format of the document file to detect when the decoy information is loaded into memory. By embedding specific beacon signatures in the decoy documents, the host-based monitoring application can detect and receive beacon signals each time the decoy documents are accessed, opened, etc. Information about the purloined document can be uploaded to the host-based monitoring application.

[0155] In yet another suitable embodiment, deception system 114 can implement one or more beacons in connection with a network intrusion detection system. A network intrusion detection system, such as Snort, can be used to detect these embedded beacons or tokens during the egress or exfiltration of the decoy document in network traffic.

[0156] In some embodiments, the decoy document itself can be used to detect inside attackers at the time of information exploitation and/or credential misuse. For example, the content of the decoy information can include a decoy login (e.g., a decoy login and password for Google Mail) and/or other credentials embedded in the document content. The bogus login to a website can be created in a decoy document and can be monitored by external approaches (e.g., using a custom script that accesses mail.google.com and parses the bait account pages to gather account activity information). Monitoring the use of decoy information by external systems (e.g., a local IT system, at Gmail, at an external bank) can be used to generate an alert that is correlated with the decoy document in which the credential was placed.

[0157] For example, if deception system 114 creates unique decoy usernames for each computer in system 100, the use of a unique decoy username can assist deception system 114 in determining which computer has been compromised, the identity of the inside attacker, etc. 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.

[0158] 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.

[0159] 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 Dec. 1, 2005, Stolfo et al., U.S. Patent Publication No. 2005/0281291 A1, published Dec. 22, 2005, and Stolfo et al., U.S. Patent Publication No. 2006/0015630 A1, published Jan. 19, 2006, which are hereby incorporated by reference herein in their entireties.

[0160] An illustrative example of the execution of an embedded active beacon in a decoy document is shown in FIG. 19. 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 1902. Window 1902 requests that the attacker allow a connection to a particular website. In response to selecting the "Allow" button or any other suitable user interface, the beacon causes a signal to be transmitted to the website (adobe-fonts.cs.columbia.edu) with information relating to the exfiltrated document and/or information relating to the attacker (as shown in FIG. 20).

[0161] 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. 19, 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.

[0162] In some embodiments, deception system 114 can instruct the legitimate user to configure the local machine to allow the one or more beacons to silently transmit signals to a remote website. For example, the first time a decoy document, such as tax document 204 of FIG. 2, is downloaded, deception system 114 can instruct the legitimate user to open the decoy document for review. In response to opening the decoy document, the application, such as Adobe Acrobat, runs a Javascript function that displays window 1902 that warns the user that the document is attempting to make a network connection with a remote server. Deception system 114 can instruct the user to configure the application to allow the beacons embedded in the decoy document to silently transmit signals to the remote website. For example, deception system 114 can instruct the user to select a "Remember this action" box and an "Allow" box such that subsequently opening the decoy document does not generate the warning message. The warning message can indicate to the inside attacker that the document is a decoy document.

[0163] 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.

[0164] 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. 21 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.

[0165] In some embodiments, beacon signals are transmitted to deception system 114, where deception system 114 provides the legitimate user with an interface showing each alert received from beacons embedded in decoy documents associated with the legitimate user. In response, the legitimate user can review particular IP addresses, review which documents are being accessed and/or misused by inside attackers, etc. Generally speaking, the legitimate user can gain an understanding of what an inside attacker may be searching for on the legitimate user's device.

[0166] In addition, deception system 114 can transmit an email notification to the legitimate user that indicates that an inside attacker may be present. As shown in FIG. 22, the notification can include information relating to the attacker, such as the IP address, the exfiltrated document, and the time that the attacker opened the document. As also shown, the notification can include count information relating to the number of times the particular decoy document has been accessed, executed, etc.

[0167] In accordance with some embodiments, decoy information with embedded beacons are implemented using a process 2300 as illustrated in FIG. 23. 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.

[0168] As shown, at 2302, 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.

[0169] 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.

[0170] At 2304, 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.

[0171] 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.

[0172] 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.

[0173] 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.

[0174] At 2306, 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 appli-

cation 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.

[0175] In accordance with some embodiments, decoy information with embedded beacons can transmit additional notifications and/or recommendations using a process 2400 as illustrated in FIG. 24.

[0176] As shown, at 2402, 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.

[0177] Alternatively, deception system 114 polls a number of servers for information to monitor decoy credential usage or any other suitable decoy information. For example, an alert component of deception system 114 can poll a number of servers to monitor credential usage, such as university authentication log servers and mail.google.com for Gmail account usage. More particularly, with regard to Gmail accounts, the alert component of deception system 114 can create custom scripts that access and parse the bait account pages to gather account activity information.

[0178] 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 2404. 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 2406, 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).

[0179] 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.

[0180] Alternatively to using beacons to transmit signals to a remote website, beacons can be associated and/or embedded with decoy information to allow a legitimate user to differentiate decoy information from actual information. As

described previously, the embedded beacon can be a portion of code that is configured to operate along with a physical mask, such as a uniquely patterned transparent screen. For example, a pattern can be generated on the display monitor in a bounded box. When the physical mask is overlaid on the displayed window containing the generated pattern, a derived word, picture, icon, or any other suitable indicia can be revealed that allows the legitimate user to discriminate between decoy information and actual information. In another example, the embedded beacon generates a pattern that is a convolution of the indicia and the physical mask allows a user to decode the pattern.

[0181] In some embodiments, multiple passive beacons can be embedded in a document that contains both actual and decoy information. When a physical mask is overlaid on the displayed window containing generated patterns for each passive beacon, indicia can be revealed that allows the legitimate user to determine which information is decoy information. For example, the indicia can provide the user with instructions on which information is decoy information.

[0182] As described above, deception system 114 can be modeled based on different levels of insider sophistication and capability. For example, some inside attackers have tools available to assist in determining whether a document is a decoy document or a legitimate document, while other inside attackers are equipped with their own observations and thoughts. Deception system 114 can be designed to confuse, deceive, and/or detect low threat level inside attackers having direct observation as the tool available, medium threat level inside attackers that have the opportunity to perform a more thorough investigation, high threat level inside attackers that have multiple tools available (e.g., super computers, access to informed people with organizational information), and/or highly privileged threat level inside attackers that may be aware that the system is baited with decoy information and that use tools to analyze, disable, and/or avoid decoy information. To do this, in some embodiments, multiple beacons or detection mechanisms can be placed in decoy documents or any other suitable decoy information, where these multiple detection mechanisms act synergistically to detect access or attempted exfiltration by an inside attacker, an external attacker, or malware and make it difficult for an attacker to avoid detection. This is sometimes referred to herein as a “web of detectors.”

[0183] In some embodiments, a decoy document generation component can be combined with a network component that monitors network traps and/or decoy traffic. For example, as described above, the decoy document generation component generates realistic documents that contain decoy credentials that are monitored for misuse and stealthy embedded beacons that signal when the document is accessed. The network component includes monitored network traps that are tied into the decoy document generation component. These network traps allow targeted credentials to be followed even after leaving the local system.

[0184] In another example, within a decoy document, the decoy document can include an embedded honeypot with a computer login account that provides no access to valuable resources and that is monitored when misused. The decoy document can also include an embedded honeypot with a banking login account that is created and monitored to entice financially motivated attackers. The decoy document can further include a network-level egress monitor that alerts whenever a marker or passive beacon, planted in the decoy docu-

ment, is detected. The decoy document can further include a host-based monitor that alerts whenever a decoy document is touched in the file system (e.g., a copy operation). The decoy document can even further include an embedded active beacon that alerts a remote server at a particular website. In turn, the website sends an email alert to the registered user that created and downloaded the decoy document.

[0185] In some embodiments, the efficacy of the generated 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 information 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 information can be measured by monitoring egress traffic or file system access. In some embodiments, the efficacy of the generated decoy information can be used to generate reports on the security of a collaborative system or any other suitable device.

[0186] 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.

[0187] 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.

[0188] In accordance with some embodiments, the techniques and mechanisms described herein can be used to measure the computer security of users, a group of users, an organization, etc. Such a measurement can be used to generate a computer security profile of the user, group of users, organization, etc. Computer security can be reflective of the likelihood a user is going to click on a link in an email from an unknown party, the likelihood a user is going to click on a link in an email relating to a popular topic (e.g., a release of a new electronic gadget) versus a non-popular topic, the likelihood a user is going to reveal personal or confidential information (e.g., such as revealing the user's Social Security number), the likelihood a user is going to infect a computer with a virus, trojan, etc. (e.g., by clicking on a virus-containing executable in an email or accessing a virus-containing Web site), etc.

[0189] In accordance with some embodiments, such measurements and/or profiles can be used to improve computer security by enabling a comparison of measurements and/or profiles before and after changes to computer security hardware, software, training, usage rules, etc.

[0190] In accordance with some embodiments, such measurements and/or profiles can be used to identify changes in usage that may indicate that a user, department, or organization has become a threat. For example, a user that becomes hostile to an organization may attempt to sabotage computer systems of the organization, steal confidential information (e.g., trade secrets, financial data, etc.), etc. As another example, a user's credentials may be stolen by a masquerader posing as the user, and that masquerader may attempt to sabotage computer systems of the organization, steal confidential information (e.g., trade secrets, financial data, etc.), etc.

[0191] A process 2500 that can be used to measure computer security, generate profiles, present statistics, and detect threats in accordance with some embodiments is illustrated in FIG. 25. As shown, after process 2500 begins at 2502, the process can make decoys and/or other non-threatening access violations accessible to users at 2504. As described above, such decoys and non-threatening access violations can be designed and implemented so that they do not in fact present a security risk, but a user accessing such decoys and non-threatening access violations indicates that the user could have caused a security risk.

[0192] For example, when a received email contains a virus-containing executable file, a user clicking on that executable file could cause the virus to be executed and installed on the user's computer—thus causing a security risk. When a decoy email containing an executable file that looks like a virus (but in fact is not a virus) is received by a user, and that user clicks on the file, the user clicking on the file indicates that the user could have caused a virus to be executed and installed and therefore the user's action presents a security risk.

[0193] These decoys and/or non-threatening access violations can be presented to any suitable users, and for any suitable periods of time. For example, a decoy can be presented to a user in an email sent to the user. As another example, a decoy can be presented to a user in search results in a document management system. As yet another example, decoys can be presented in a file folder on a computer disk drive that is marked as confidential.

[0194] Next, at 2506, process 2500 can maintain statistics on security violations and non-violations of users. For example, process 2500 can monitor the time, duration, number of uses, context (such as files opened, amount of data processed (e.g., per hour CPU usage, egress data flows per hour, etc.), etc.), and any other suitable characteristics of usage of decoys, non-threatening violations, permitted applications, etc. by specific users. Any suitable statistics can be maintained, and any suitable techniques and/or mechanisms for gathering and maintaining these statistics can be used in some embodiments. For example, such statistics can include histograms, models, etc. These statistics can be generated at the user level, group level, organization level, etc. In some embodiments, statistics can be calculated at each host to reduce the data acquisition necessary, and reduce the need to mix data from multiple users. In some embodiments, these statistics can be continuously updated, periodically updated, and/or updated at any suitable point(s) in time.

[0195] In some embodiments, statistic can be kept in an anonymous fashion so as to preserve privacy of users when desired or necessary. For example, a user's name or other identifier can be hashed and the hash can be used to identify the source of corresponding statistics.

[0196] At 2508, any suitable portion of the statistics can be presented to an administrator and/or any other suitable user. For example, these statistics can be presented as a score for each user of the user's security risk. More particularly, for example, a first user who repeatedly accesses decoys may present a higher security risk than a second user and thus may have a worse security risk score. Such scores can be presented as a listed on the user names and scores sorted by score with the worst score at the top and the best score at the bottom. This can clearly indicate to the administrator which users are the biggest security risk so that those users can be more carefully monitored. As another example, these statistics can be presented as histograms so that the scores of an organization's users can be better understood. In some embodiments, these statistics can be presented in a dashboard.

[0197] Process 2500 can then determine if security violations of users exceed one or more thresholds at 2510. Any suitable technique or mechanism can be used to determine if a user's security violations exceed a threshold. For example, in some embodiments, statistics of each user can be compared to average, median, and/or clusters of statistics to determine if a user is outside a given range from the average, median, and/or clusters. As another example, in some embodiments statistics of users can be compared to threshold values (e.g., a threshold score) to determine if a user's score is below some value. As yet another example, statistics of a user can be monitored to determine if the statistics rapidly change. More particularly, for example, the statistics can be monitored to determine if a user newly accesses areas of a computer network, application, files, etc. that the user does not usually access. As still another example, statistics can include profiles of users, and profiles that are "distant" form some cluster of similar profiles can be determined as being more "suspicious" especially if the applications used and measured in the profile are deemed to be "sensitive." Such suspicious profiles can be determined to exceed a threshold.

[0198] If it is determined that one or more security violations of one or more users exceed a threshold, at 2512, process 2500 can branch to 2514 to generate an alert. Any suitable alert can be generated. For example, an alert can be generated in a dashboard of an administrator. As another example, an alert can be generated as an email sent to an administrator. As yet another example, an alert can be generated as a log entry.

[0199] If it is determined that no violations of a user exceed a threshold, or after generating an alert, process 2500 can loop back to 2504.

[0200] Accordingly, methods, systems, and media for providing trap-based defenses using decoy information are provided.

[0201] 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 measuring computer security, comprising: making at least one of decoys and non-threatening access violations accessible to a first user using a computer programmed to do so; maintaining statistics on security violations and non-violations of the first user using a computer programmed to do so; and presenting the statistics on a display.
2. The method of claim 1, further comprising: determining if the security violations exceed a threshold; and generating an alert if the security violations exceed the threshold.
3. The method of claim 1, wherein a decoy is made accessible to the first user and the decoy is contained in an email.
4. The method of claim 1, wherein a decoy is made accessible to the first user and the decoy is a file presented in a file folder.
5. The method of claim 1, wherein the statistics are maintained as one or more histograms.
6. The method of claim 1, further comprising: making at least one of decoys and non-threatening access violations accessible to a second user using a computer programmed to do so; maintaining statistics on security violations and non-violations of the second user using a computer programmed to do so; and presenting the statistics on a display.
7. The method of claim 6, further comprising comparing the statistics of the first user to the statistics of the second user.
8. A system for measuring computer security, comprising: a processor that: makes at least one of decoys and non-threatening access violations accessible to a first user; maintains statistics on security violations and non-violations of the first user; and presents the statistics on a display.
9. The system of claim 8, wherein the processor also: determines if the security violations exceed a threshold; and generates an alert if the security violations exceed the threshold.
10. The system of claim 8, wherein a decoy is made accessible to the first user and the decoy is contained in an email.
11. The system of claim 8, wherein a decoy is made accessible to the first user and the decoy is a file presented in a file folder.
12. The system of claim 8, wherein the statistics are maintained as one or more histograms.
13. The system of claim 8, wherein the processor also: makes at least one of decoys and non-threatening access violations accessible to a second user; maintains statistics on security violations and non-violations of the second user; and presents the statistics on a display.
14. The system of claim 13, wherein the processor also compares the statistics of the first user to the statistics of the second user.
15. A non-transitory computer-readable medium containing computer-executable instructions that, when executed by a processor, cause the processor to perform a method for measuring computer security, the method comprising: making at least one of decoys and non-threatening access violations accessible to a first user; maintaining statistics on security violations and non-violations of the first user; and presenting the statistics on a display.
16. The non-transitory computer-readable medium of claim 15, wherein the method further comprises: determining if the security violations exceed a threshold; and generating an alert if the security violations exceed the threshold.
17. The non-transitory computer-readable medium of claim 15, wherein a decoy is made accessible to the first user and the decoy is contained in an email.
18. The non-transitory computer-readable medium of claim 15, wherein a decoy is made accessible to the first user and the decoy is a file presented in a file folder.
19. The non-transitory computer-readable medium of claim 15, wherein the statistics are maintained as one or more histograms.
20. The non-transitory computer-readable medium of claim 15, wherein the method further comprises: making at least one of decoys and non-threatening access violations accessible to a second user; maintaining statistics on security violations and non-violations of the second user; and presenting the statistics on a display.
21. The non-transitory computer-readable medium of claim 20, wherein the method further comprises comparing the statistics of the first user to the statistics of the second user.

* * * * *