US00RE47558E

(19) **United States**

(12) **Reissued Patent**

Gryaznov

(10) **Patent Number: US RE47,558 E**

(45) **Date of Reissued Patent: Aug. 6, 2019**

(54) **SYSTEM, METHOD, AND COMPUTER PROGRAM PRODUCT FOR AUTOMATICALLY IDENTIFYING POTENTIALLY UNWANTED DATA AS UNWANTED**

(71) Applicant: **McAfee, LLC**, Plano, TX (US)

(72) Inventor: **Dmitry O. Gryaznov**, Portland, OR (US)

(73) Assignee: **McAfee, LLC**, Santa Clara, CA (US)

(21) Appl. No.: **14/527,749**

(22) Filed: **Oct. 29, 2014**

Related U.S. Patent Documents

Reissue of:

(64) Patent No.: **8,301,904**
Issued: **Oct. 30, 2012**
Appl. No.: **12/144,967**
Filed: **Jun. 24, 2008**

(51) **Int. Cl.**

G06F 21/55	(2013.01)
G06F 21/56	(2013.01)
H04L 29/06	(2006.01)
G06F 11/30	(2006.01)

(52) **U.S. Cl.**
CPC ***G06F 21/552*** (2013.01); ***G06F 21/56*** (2013.01); ***H04L 63/1408*** (2013.01)

(58) **Field of Classification Search**
CPC H04L 63/1408; G06F 21/552; G06F 21/56
USPC 726/22, 23, 24; 713/188
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,697,948	B1	2/2004	Rabin	
6,708,212	B2	3/2004	Porras et al.	
6,981,155	B1	12/2005	Lyle et al.	
7,095,716	B1	8/2006	Ke et al.	
7,409,712	B1 *	8/2008	Brooks et al.	726/22
7,512,977	B2	3/2009	Cook et al.	
7,555,777	B2	6/2009	Swimmer et al.	
7,694,150	B1	4/2010	Kirby	
7,752,667	B2 *	7/2010	Challener et al.	726/24
7,802,303	B1	9/2010	Zhao et al.	
7,912,872	B2	3/2011	Bayiates	
7,945,787	B2	5/2011	Gassoway	
8,301,904	B1	10/2012	Gryaznov	
8,590,039	B1	11/2013	Muttick et al.	
8,627,461	B2	1/2014	Barton et al.	
8,719,939	B2	5/2014	Krasser et al.	
9,106,688	B2	8/2015	Muttik et al.	

(Continued)

FOREIGN PATENT DOCUMENTS

WO	WO 2008/089626	7/2008
WO	WO 2011/082084	7/2011

OTHER PUBLICATIONS

"chroot(2)—Linux man page" http://linux.die.net/man/2/chroot. Downloaded on Feb. 27, 2008 from—http://linux.die.net/man/2/chroot—pp. 1-2.

(Continued)

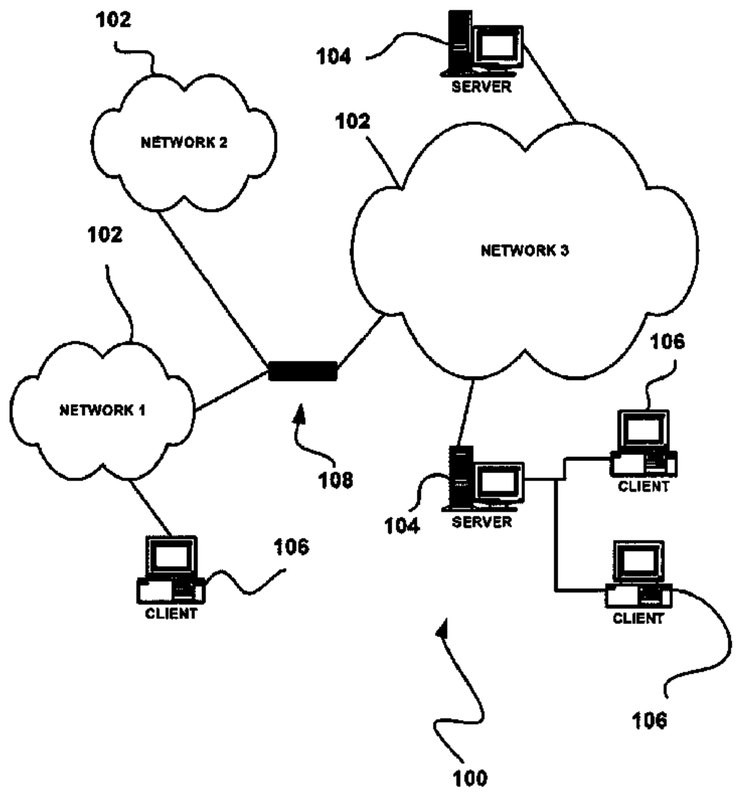
Primary Examiner — Matthew E Heneghan

(74) *Attorney, Agent, or Firm* — Patent Capital Group

(57) **ABSTRACT**

A system, method, and computer program product are provided for automatically identifying potentially unwanted data as unwanted. In use, data determined to be potentially unwanted (e.g. potentially malicious) is received. Additionally, the data is automatically identified as unwanted (e.g. malicious). Furthermore, the data is stored for use in detecting unwanted data (e.g. malicious data).

45 Claims, 6 Drawing Sheets



(56) **References Cited**

U.S. PATENT DOCUMENTS

9,306,796	B1	4/2016	Muttik et al.	
2004/0042416	A1	3/2004	Ngo et al.	
2004/0044912	A1*	3/2004	Connary	H04L 43/045 726/23
2004/0054925	A1*	3/2004	Etheridge	H04L 63/1458 726/22
2004/0073810	A1	4/2004	Dettinger et al.	
2004/0078592	A1*	4/2004	Fagone	H04L 63/14 726/22
2004/0123117	A1	6/2004	Berger	
2004/0203589	A1*	10/2004	Wang et al.	455/410
2004/0255163	A1	12/2004	Swimmer et al.	
2005/0015455	A1	1/2005	Liu	
2005/0027818	A1	2/2005	Friedman et al.	
2005/0065899	A1	3/2005	Cong et al.	
2005/0177868	A1	8/2005	Kwan	
2005/0262567	A1	11/2005	Carmona	
2005/0262576	A1	11/2005	Gassoway	
2006/0036693	A1	2/2006	Hulten et al.	
2006/0070130	A1*	3/2006	Costea et al.	726/24
2006/0137012	A1*	6/2006	Aaron	G06F 21/564 726/24
2006/0150256	A1	7/2006	Fanton et al.	
2006/0230452	A1	10/2006	Field	
2006/0242245	A1	10/2006	Christensen	
2007/0016953	A1	1/2007	Morris et al.	
2007/0028304	A1	2/2007	Brennan	
2007/0073660	A1	3/2007	Quinlan	
2007/0079379	A1*	4/2007	Sprosts et al.	726/24
2007/0226804	A1	9/2007	Somkiran et al.	
2007/0240217	A1	10/2007	Tuvell et al.	
2007/0240220	A1	10/2007	Tuvell et al.	
2007/0261112	A1	11/2007	Todd et al.	
2008/0126779	A1	5/2008	Smith	
2008/0127336	A1	5/2008	Sun et al.	
2008/0141373	A1	6/2008	Fossen et al.	
2008/0168533	A1	7/2008	Ozaki et al.	
2008/0196099	A1	8/2008	Shastri	
2008/0295177	A1	11/2008	Dettinger et al.	
2008/0313738	A1	12/2008	Enderby	
2009/0044024	A1	2/2009	Oberheide et al.	
2009/0064329	A1	3/2009	Okumura et al.	
2009/0064337	A1	3/2009	Chien	
2009/0083852	A1	3/2009	Kuo et al.	
2009/0088133	A1	4/2009	Orlassino	
2009/0097661	A1	4/2009	Orsini et al.	
2009/0254992	A1	10/2009	Schultz et al.	
2010/0031358	A1	2/2010	Elovici et al.	
2011/0047618	A1	2/2011	Evans et al.	
2011/0138465	A1	6/2011	Franklin et al.	
2011/0162070	A1	6/2011	Krasser et al.	
2011/0197177	A1	8/2011	Mony	
2012/0084859	A1	4/2012	Radinsky et al.	
2013/0276106	A1	10/2013	Barton et al.	
2013/0276120	A1	10/2013	Dalcher et al.	
2014/0053263	A1	2/2014	Muttik et al.	
2016/0036832	A1	2/2016	Muttik et al.	
2016/0261620	A1	9/2016	Muttik et al.	

OTHER PUBLICATIONS

Non-Final Office Action, dated Dec. 29, 2011 for U.S. Appl. No. 11/946,777.
Advisory Action dated 5, 2012 for U.S. Appl. No. 12/398,073 (3 pages), July.
Non-Final Office Action dated Mar. 13, 2012 for U.S. Appl. No. 12/693,765 (13 pages).
Non-Final Office Action dated Mar. 15, 2012 for U.S. Appl. No. 12/144,967 (8 pages).
Non-Final Office Action in U.S. Appl. No. 12/144,967 dated Mar. 3, 2011 (8 pages).
U.S. Appl. No. 14/063,813 which was filed Oct. 25, 2013 (24 pages).
Non-Final Office Action dated Feb. 11, 2015 for U.S. Appl. No. 14/063,813 (18 pages).
Notice of Allowance dated Apr. 16, 2015 for U.S. Appl. No. 14/063,813 (10 pages).
U.S. Appl. No. 14/823,855 which was filed Aug. 11, 2015 (21 pages).
Non-Final Office Action dated Apr. 22, 2016 for U.S. Appl. No. 14/823,855 (11 pages).
Notice of Allowance dated Nov. 23, 2016 for U.S. Appl. No. 14/823,855 (11 pages).
U.S. Appl. No. 15/070,051 which was filed Mar. 15, 2016 (18 pages).
Chouchane, Mohamed R., Andrew Walenstein, and Arun Lakhotia. "Statistical signatures for fast filtering of instruction-substituting metamorphic malware." Proceedings of the 2007 ACM workshop on Recurring malcode. ACM, 2007 (7 pages), Retrieved from internet on Mar. 8, 2017 at https://webcache.googleusercontent.com/search?q=cache:RcgpFElyJe0J:https://cs.columbusstate.edu/cae-ia/facultypapers/chouchane/2007-chouchane-walenstein-lakhotia.pdf+&cd=1&hl=en&ct=clnk&gl=us.
Hu, Guoning, and Deepak Venugopal. "A malware signature extraction and detection method applied to mobile networks." Performance, Computing, and Communications Conference, 2007. IPCCC 2007. IEEE international. IEEE, 2007.
Xu, J-Y., et al, "Polymorphic malicious executable scanner by API sequence analysis." Hybrid Intelligent Systems, 2004. HIS'04. Fourth International Conference on IEEE. 2004.
Notice of Allowance received for U.S. Appl. No. 11/946,777, dated Jul. 19, 2013 (12 pages).
Non-Final Office Action received for U.S. Appl. No. 11/946,777, dated Feb. 1, 2013, 5 pages.
Final Office Action from U.S. Appl. No. 12/050,432 dated Jun. 21, 2012 (9 pages).
Notice of Allowance from U.S. Appl. No. 12/050,432 dated Dec. 16, 2015 (5 pages).
Advisory Action dated Jul. 29, 2011 in U.S. Appl. No. 12/050,432 (4 pages).
Non Final Office Action dated Sep. 11, 2013 in U.S. Appl. No. 12/131,383 (29 pages).
Final Office Action dated Mar. 7, 2014 for U.S. Appl. No. 12/131,383 (32 pages).
Non-Final Office Action dated Feb. 15, 2013 for U.S. Appl. No. 12/398,073 (12 pages).
Notice of Allowance dated Jun. 24, 2013 for U.S. Appl. No. 12/398,073 (10 pages).
Notice of Allowance dated Aug. 30, 2013 for U.S. Appl. No. 12/398,073 (10 pages).
Final Office Action received for U.S. Appl. No. 12/144,967 dated Aug. 17, 2011, 8 pages.
Notice of Allowance from U.S. Appl. No. 12/144,967 dated Aug. 17, 2012 (7 pages).
International Preliminary Report received for PCT Patent Application No. PCT/US2010/061889, dated Jul. 4, 2012, 4 pages.
International Search Report and Written Opinion received for PCT Patent Application No. PCT/US2010/061889, dated Aug. 29, 2011, 6 pages.
Final Office Action dated Jun. 28, 2012 for U.S. Appl. No. 12/131,383 (27 pages).
Office Action for Australian Patent Application No. 2010336989, dated Jun. 21, 2013, 3 pages.
Korean Intellectual Property Office Notice of Grounds for Refusal in Korean Patent Application No. 10-2012-7020220, dated Sep. 23, 2013, 15 pages of Office Action including 5 pages of English Translation.
U.S. Appl. No. 12/398,073, filed Mar. 4, 2009 (24 pages).
U.S. Appl. No. 12/693,765, filed Jan. 26, 2010 (16 pages).
U.S. Appl. No. 12/144,967 which was filed Jun. 24, 2008 (31 pages).
Provisional U.S. Appl. No. 61/291,568 which was filed Dec. 31, 2009 (13 pages).
U.S. Appl. No. 12/131,383, which was filed Jun. 2, 2008.
U.S. Appl. No. 11/946,777, which was filed Nov. 28, 2007.
U.S. Appl. No. 12/111,846, which was filed Apr. 29, 2008.

(56) **References Cited**

OTHER PUBLICATIONS

"VMWare DiskMount Utility: User's Manual", http://www.vmware.com/pdf/VMwareDiskMount.pdf, 1998-2005, Revision Apr. 8, 2005, VMWare, Inc., 6 pages.
Wolf, Chris, Column: "Virtual Server 2005 R2 SP1 Treasures: VHD Mount", Jun. 2007, Microsoft Certified Professional Magazine Online, Downloaded on Feb. 27, 2008 from—http://mcpmag.com/columns/article.asp?EditorialsID=1793—pp. 1-5.
"Linux/Unix Command: chroot", Downloaded on Feb. 27, 2008 from—http://linux.about.com/library/cmd/blcmd12_chroot. htm—pp. 1-3.
Non-Final Office Action Summary from U.S. Appl. No. 11/946,777 dated Jan. 5, 2011.
Christodorescu, Miha et al. "Testing Malware Detectors", In the Proceedings of the ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA '04), vol. 29, Issue 4, Jul. 11-14, 2004, Boston Massachusetts, 11 pages.
"Blacklist," Wikipedia, last modified Jun. 5, 2008, http://en.wikipedia.org/Wiki/Blacklist.
Office Action Summary from U.S. Appl. No. 12/111,846 dated Jun. 24, 2011.
Office Action Summary from U.S. Appl. No. 12/131,383 dated Jun. 24, 2011.
An Architecture for Generating Semantics-Aware Signatures; Vinod Yegneswaran, Jonathon T. Giffin, Paul Barford, Somesh Jha; Appeared in Proceedings of Usenix Security Symposium 2005, year 2005, all pages.
U.S. Appl. No. 12/050,432, which was filed Mar. 18, 2008.
Offce Action Summary from U.S. Appl. No. 11/946,777 dated Jun. 13, 2011.
Office Action Summary from U.S. Appl. No. 12/050,432 dated Oct. 6, 2010.
Office Action Summary from U.S. Appl. No. 12/050,432 dated May 13, 2011.
Non-Final Office Action dated Mar. 12, 2012 for U.S. Appl. No. 12/050,432.
Final Office Action dated Oct. 17, 2011 for U.S. Appl. No. 12/131,383.
Non-Final Office Action dated Mar. 6, 2012 for U.S. Appl. No. 12/131,383.
U.S. Appl. No. 12/398,073, filed Mar. 4, 2009.
Non-Final Office Action dated Oct. 4, 2011 for U.S. Appl. No. 12/398,073.
Final Office Action dated Apr. 12, 2012 for U.S. Appl. No. 12/398,073.
Final Office Action, dated Dec. 29, 2011 for U.S. Appl. No. 11/946,777.

* cited by examiner

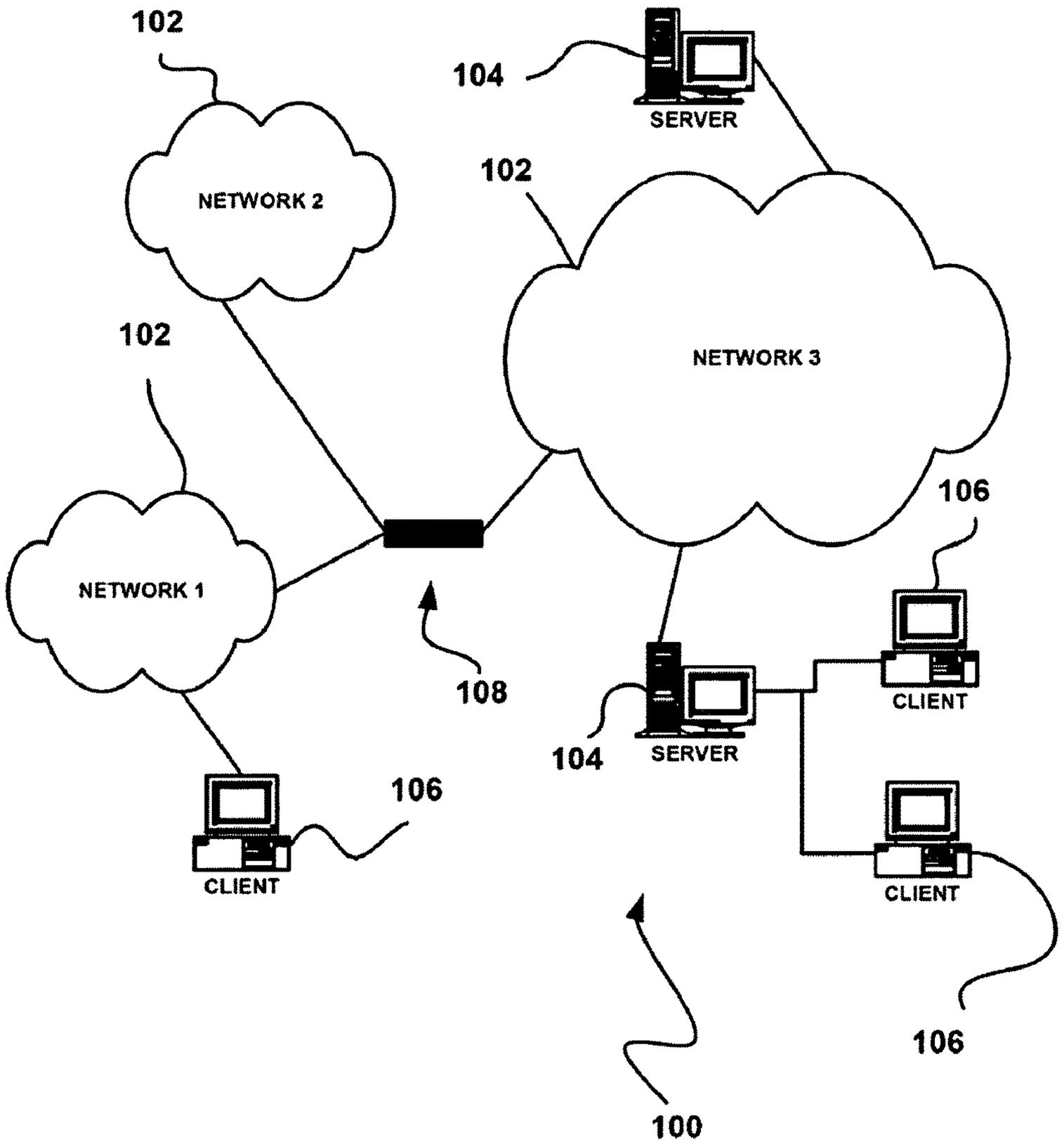

102
NETWORK 2
102
NETWORK 1
104
SERVER
102
NETWORK 3
106
108
104
SERVER
CLIENT
106
CLIENT
CLIENT
106
100

FIGURE 1

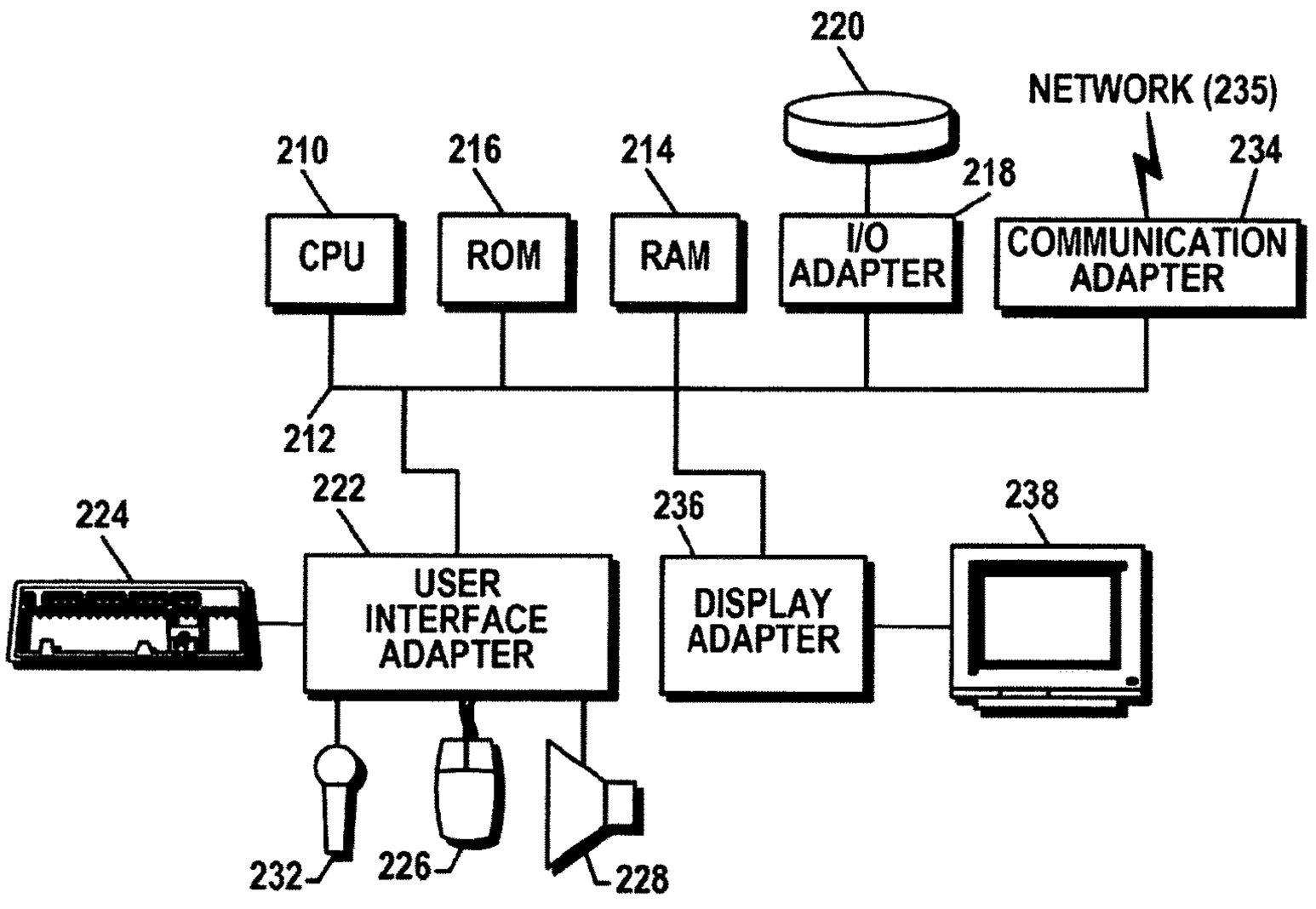

220
NETWORK (235)
210
216
214
218
234
CPU
ROM
RAM
I/O ADAPTER
COMMUNICATION ADAPTER
212
222
236
238
224
USER INTERFACE ADAPTER
DISPLAY ADAPTER
232
226
228

FIGURE 2

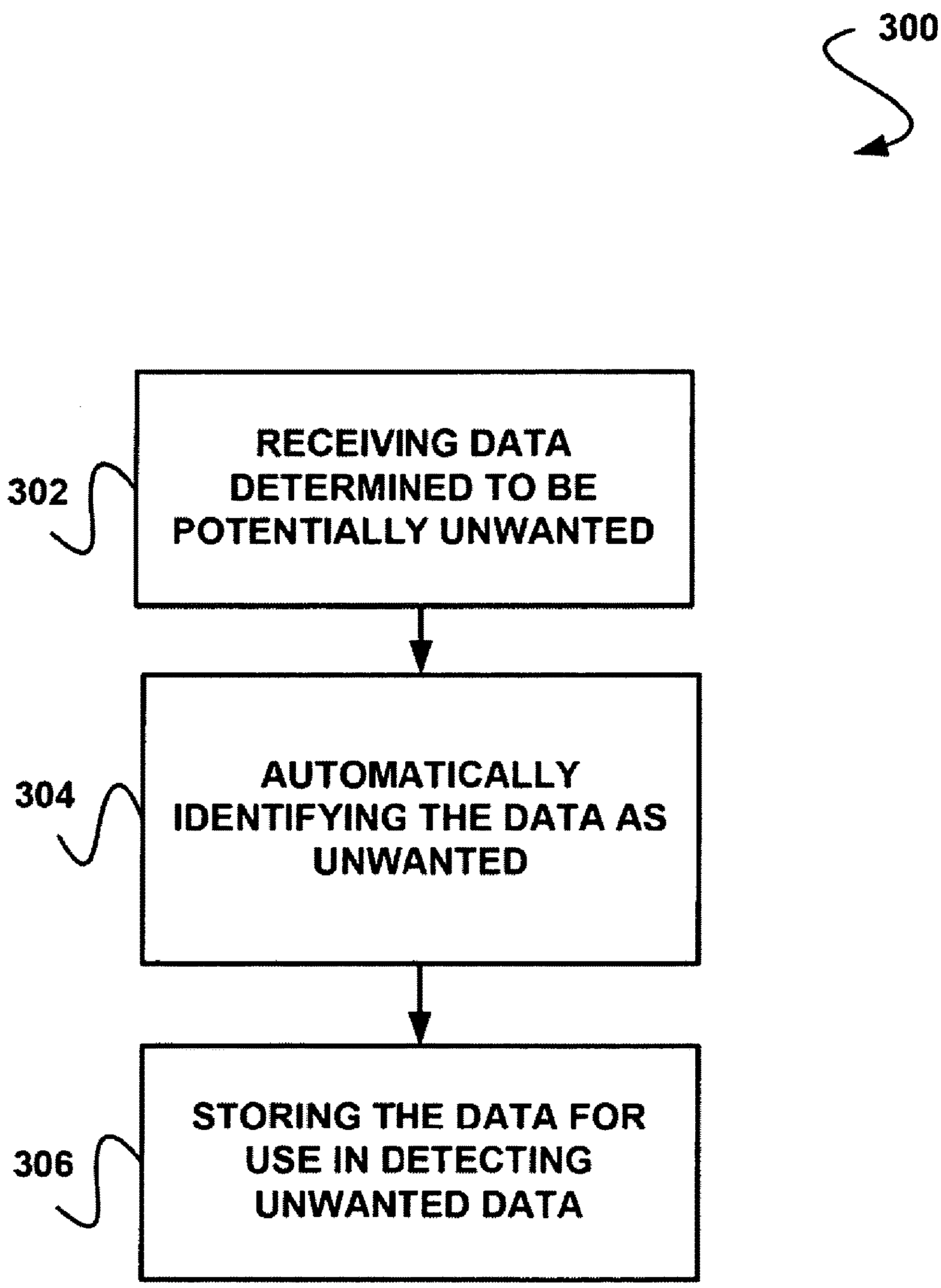

300
302
RECEIVING DATA DETERMINED TO BE POTENTIALLY UNWANTED
304
AUTOMATICALLY IDENTIFYING THE DATA AS UNWANTED
306
STORING THE DATA FOR USE IN DETECTING UNWANTED DATA

FIGURE 3

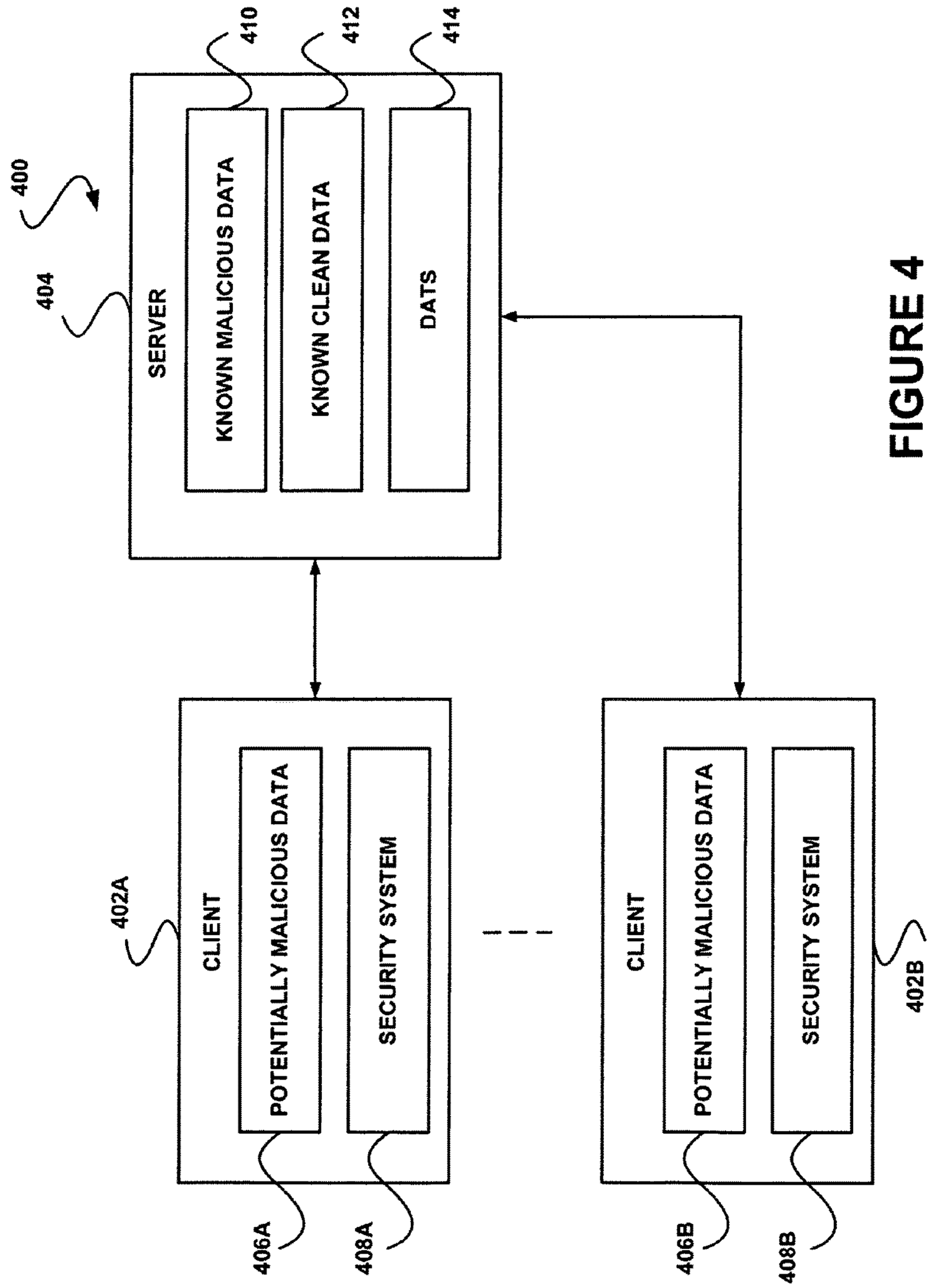

400
404
SERVER
410
412
414
KNOWN MALICIOUS DATA
KNOWN CLEAN DATA
DATS
402A
CLIENT
POTENTIALLY MALICIOUS DATA
SECURITY SYSTEM
406A
408A
402B
CLIENT
POTENTIALLY MALICIOUS DATA
SECURITY SYSTEM
406B
408B

FIGURE 4

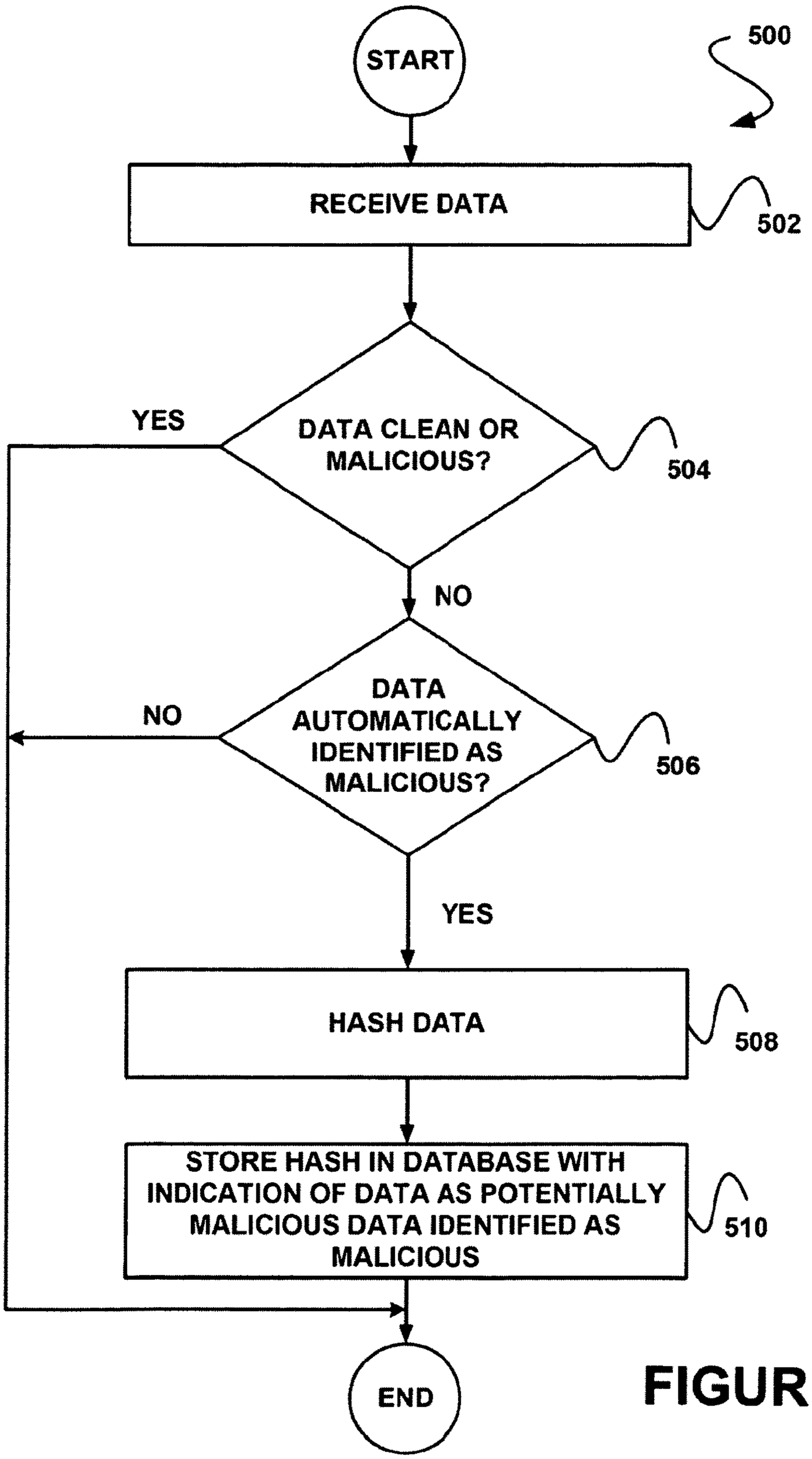

START
500
RECEIVE DATA
502
DATA CLEAN OR MALICIOUS?
YES
504
NO
DATA AUTOMATICALLY IDENTIFIED AS MALICIOUS?
NO
506
YES
HASH DATA
508
STORE HASH IN DATABASE WITH INDICATION OF DATA AS POTENTIALLY MALICIOUS DATA IDENTIFIED AS MALICIOUS
510
END

FIGURE 5

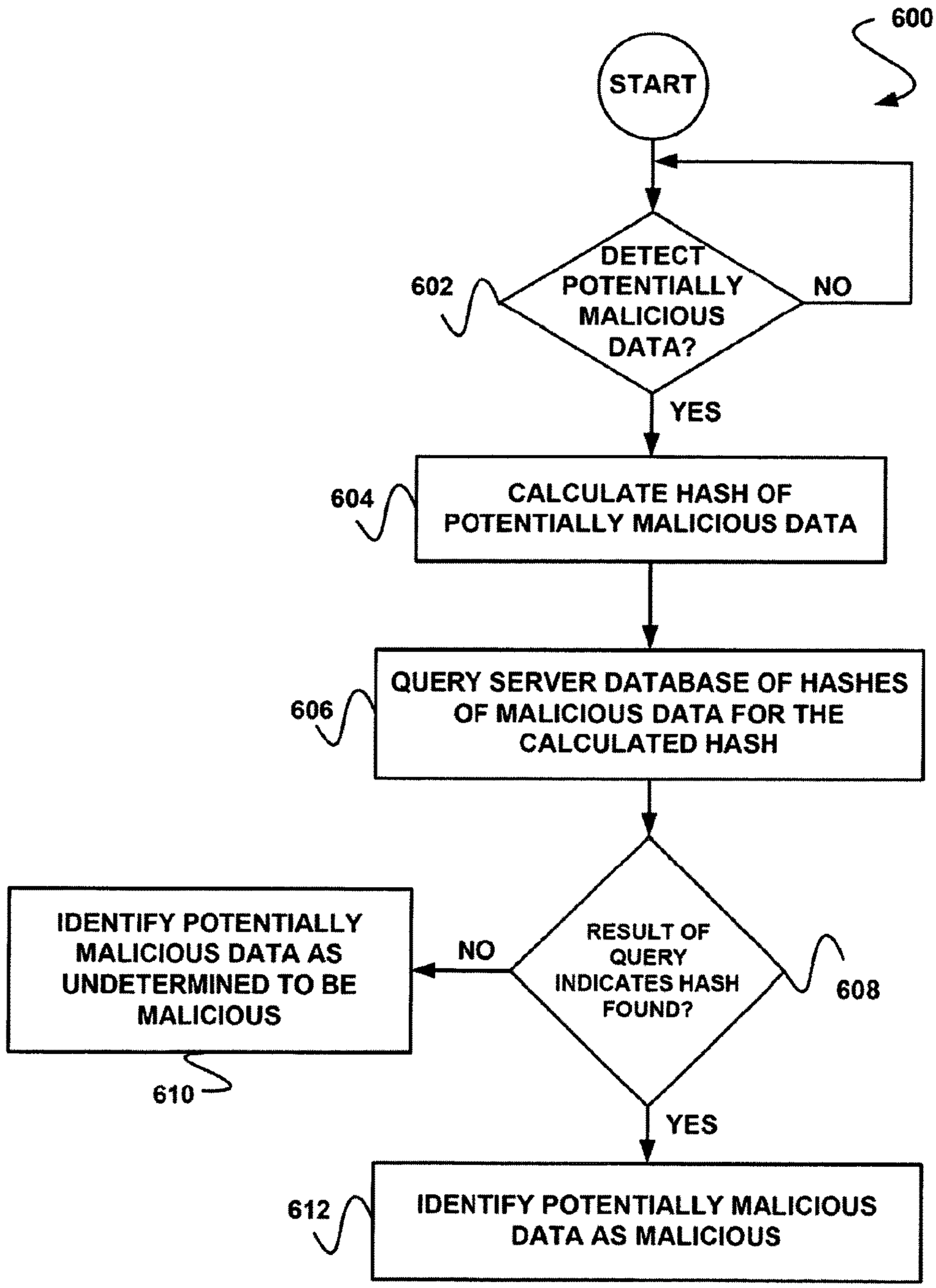

600
START
602
DETECT POTENTIALLY MALICIOUS DATA?
NO
YES
604
CALCULATE HASH OF POTENTIALLY MALICIOUS DATA
606
QUERY SERVER DATABASE OF HASHES OF MALICIOUS DATA FOR THE CALCULATED HASH
608
RESULT OF QUERY INDICATES HASH FOUND?
NO
IDENTIFY POTENTIALLY MALICIOUS DATA AS UNDETERMINED TO BE MALICIOUS
610
YES
612
IDENTIFY POTENTIALLY MALICIOUS DATA AS MALICIOUS

FIGURE 6

SYSTEM, METHOD, AND COMPUTER PROGRAM PRODUCT FOR AUTOMATICALLY IDENTIFYING POTENTIALLY UNWANTED DATA AS UNWANTED

Matter enclosed in heavy brackets [ ] appears in the original patent but forms no part of this reissue specification; matter printed in italics indicates the additions made by reissue; a claim printed with strikethrough indicates that the claim was canceled, disclaimed, or held invalid by a prior post-patent action or proceeding.

CROSS-REFERENCE TO RELATED APPLICATION

This application is a reissue application of U.S. Pat. No. 8,301,904, entitled "SYSTEM, METHOD, AND COMPUTER PROGRAM PRODUCT FOR AUTOMATICALLY IDENTIFYING POTENTIALLY UNWANTED DATA AS UNWANTED," which issued on Oct. 30, 2012, from U.S. application Ser. No. 12/144,967, filed on Jun. 24, 2008.

FIELD OF THE INVENTION

The present invention relates to security systems, and more particularly to identifying unwanted data.

BACKGROUND

Security systems have traditionally been concerned with identifying unwanted (e.g., malicious) data and acting in response thereto. For example, data which is undetermined to be malicious may be communicated to a security system, and the data may further be analyzed by the security system for determining whether the data is malicious. However, traditional techniques for determining whether data is malicious have generally exhibited various limitations.

For example, security systems that determine whether data is malicious are oftentimes in communication with multiple other devices, and therefore conventionally receive numerous requests to determine whether data is malicious from such devices. When numerous requests are received in this manner, significant delays by the security systems in determining whether the data is malicious and responding to the devices based on the determinations generally exist. Further, the responses generated by the security systems based on such determinations are customarily formed as updates to security systems installed on the devices. However, many times the devices themselves delay installation of the updates when such updates are available from the security systems, thus resulting in a delayed identification by the devices of whether data is in fact malicious.

There is thus a need for overcoming these and/or other issues associated with the prior art.

SUMMARY

A system, method, and computer program product are provided for automatically identifying potentially unwanted data as unwanted. In use, data determined to be potentially unwanted (e.g. potentially malicious) is received. Additionally, the data is automatically identified as unwanted (e.g. malicious). Furthermore, the data is stored for use in detecting unwanted data (e.g. malicious data).

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** illustrates a network architecture, in accordance with one embodiment.

FIG. **2** shows a representative hardware environment that may be associated with the servers and/or clients of FIG. **1**, in accordance with one embodiment.

FIG. **3** shows a method for automatically identifying potentially unwanted (e.g. potentially malicious) data as unwanted (e.g. malicious), in accordance with one embodiment.

FIG. **4** shows a system for automatically identifying potentially unwanted (e.g. potentially malicious) data as unwanted (e.g. malicious), in accordance with another embodiment.

FIG. **5** shows a method for storing a hash of data with an indication of whether the data is potentially malicious or potentially clean, in accordance with yet another embodiment.

FIG. **6** shows a method for querying a database of hashes for identifying potentially malicious data as malicious, in accordance with still yet another embodiment.

DETAILED DESCRIPTION

FIG. **1** illustrates a network architecture **100**, in accordance with one embodiment. As shown, a plurality of networks **102** is provided. In the context of the present network architecture **100**, the networks **102** may each take any form including, but not limited to a local area network (LAN), a wireless network, a wide area network (WAN) such as the Internet, peer-to-peer network, etc.

Coupled to the networks **102** are servers **104** which are capable of communicating over the networks **102**. Also coupled to the networks **102** and the servers **104** is a plurality of clients **106**. Such servers **104** and/or clients **106** may each include a desktop computer, lap-top computer, hand-held computer, mobile phone, personal digital assistant (PDA), peripheral (e.g., printer, etc.), any component of a computer, and/or any other type of logic. In order to facilitate communication among the networks **102**, at least one gateway **108** is optionally coupled therebetween.

FIG. **2** shows a representative hardware environment that may be associated with the servers **104** and/or clients **106** of FIG. **1**, in accordance with one embodiment. Such figure illustrates a typical hardware configuration of a workstation in accordance with one embodiment having a central processing unit **210**, such as a microprocessor, and a number of other units interconnected via a system bus **212**.

The workstation shown in FIG. **2** includes a Random Access Memory (RAM) **214**, Read Only Memory (ROM) **216**, an I/O adapter **218** for connecting peripheral devices such as disk storage units **220** to the bus **212**, a user interface adapter **222** for connecting a keyboard **224**, a mouse **226**, a speaker **228**, a microphone **232**, and/or other user interface devices such as a touch screen (not shown) to the bus **212**, communication adapter **234** for connecting the workstation to a communication network **235** (e.g., a data processing network) and a display adapter **236** for connecting the bus **212** to a display device **238**.

The workstation may have resident thereon any desired operating system. It will be appreciated that an embodiment may also be implemented on platforms and operating systems other than those mentioned. One embodiment may be written using JAVA, C, and/or C++ language, or other programming languages, along with an object oriented programming methodology. Object oriented programming (OOP) has become increasingly used to develop complex applications.

Of course, the various embodiments set forth herein may be implemented utilizing hardware, software, or any desired

combination thereof. For that matter, any type of logic may be utilized which is capable of implementing the various functionality set forth herein.

FIG. **3** shows a method **300** for automatically identifying potentially unwanted (e.g. potentially malicious) data as unwanted (e.g. malicious), in accordance with one embodiment. As an option, the method **300** may be carried out in the context of the architecture and environment of FIGS. **1** and/or **2**. Of course, however, the method **300** may be carried out in any desired environment.

As shown in operation **302**, data determined to be potentially unwanted (e.g. potentially malicious) is received. In the context of the present description, the data determined to be potentially unwanted may include any data for which it is unknown whether such data is unwanted (e.g. malicious). Thus, in one embodiment, the data may be determined to be unwanted by determining that it is unknown whether the data is unwanted. It should be noted that such data may include any code, application, file, electronic message, process, thread, etc. that is potentially unwanted.

In another embodiment, it may be determined that it is unknown whether the data is unwanted based on an analysis of the data. For example, it may be determined that it is unknown whether the data is unwanted by determining that the data does not match known wanted data (e.g. data predetermined to be wanted, whitelisted data, etc.) and that the data does not match known unwanted data (e.g. data predetermined to be unwanted, blacklisted data, etc.). To this end, the data may be compared to the known wanted data and the known unwanted data for determining whether it is unknown that the data is unwanted.

As another example, the potentially unwanted data may not necessarily match a hash, signature, etc. of known unwanted data. As another example, the potentially unwanted data may not necessarily match a hash, signature, etc. of known wanted data. Such data may be determined to be potentially unwanted based on a scan of the data (e.g., against signatures of known wanted data and/or known unwanted data, etc.), as an option.

In yet another embodiment, the data may be determined to be potentially unwanted if it is determined that the data is suspicious based on an analysis thereof. For example, the data may be determined to have one or more characteristics of malware based on the analysis. In another example, the data may be determined to be a possible new variant of existing malware. To this end, the potentially unwanted data may include data that is determined to potentially include malware, spyware, adware, etc.

Additionally, in one embodiment, the data may be determined to be potentially unwanted based on monitoring performed with respect to the data. For example, the monitoring may include identifying the data (e.g. based on operations performed in association with the data, etc.) and performing an analysis of the data, such as the analysis described above for example. Optionally, the monitoring may be of an electronic messaging application [e.g. electronic mail (email) messaging application], a file transfer protocol (FTP), at least one web site, etc.

In another embodiment, the data may be determined to be potentially unwanted based on a heuristic analysis. In yet another embodiment, the data may be determined to be potentially unwanted based on a behavioral analysis. In yet another embodiment, the data may be determined to be potentially unwanted based on scanning performed on the data. Of course, however, data may be determined to be potentially unwanted in any desired manner.

Further, the data may be determined to be potentially unwanted by a remote source. As another option, the data determined to be potentially unwanted may be received from such remote source. In one embodiment, such data may be automatically received based on the monitoring described above. Just by way of example, the remote device may automatically transmit the data in response to a determination that the data is potentially unwanted (e.g. that it is unknown whether such data is unwanted, etc.).

As an option, the data determined to be potentially unwanted may be received by a server. In one embodiment, the server may be utilized by a security vendor. Such security vendor may optionally provide known wanted data and/or known unwanted data (e.g. via updates, etc.) to a plurality of client devices, such that the client devices may utilize the known wanted data and/or known unwanted data for determining whether data is wanted and/or unwanted, respectively. To this end, the server may optionally receive the data determined to be potentially unwanted for analysis purposes, such as for determining whether the data is wanted or unwanted. Further, based on the determination, the server may be utilized to provide an indication of the determination (e.g. via an update, etc.) to a source from which the data was received and/or to any other desired device.

Moreover, as shown in operation **304**, the data is automatically identified as unwanted (e.g. malicious). In one embodiment, automatically identifying the data as unwanted may include any determination that the data is unwanted which does not necessarily rely on an analysis of the data. For example, the data may be automatically identified as unwanted without necessarily scanning the data, comparing the data to known wanted data and/or known unwanted data, etc.

In another embodiment, the data may be automatically identified as unwanted based on at least one source from which the data is received. As an option, the data may be automatically identified as unwanted based on a type of the source from which the data is received. For example, if the source includes a security vendor, a multi-scanner service, a honeypot, etc., the data may be automatically identified as unwanted.

As another option, the data may be automatically identified as unwanted if it is determined that other data previously received from the source (e.g. received previous to that received in operation **302**) includes known unwanted data. For example, if other data previously received from the source was determined to be unwanted, the data received in operation **302** may be automatically identified as unwanted. As another example, if a predefined threshold amount (e.g. percentage, etc.) of data previously received from the source was determined to be unwanted, the data received in operation **302** may be automatically identified as unwanted. In this way, the data may be automatically identified as unwanted if potentially unwanted data received from such source was determined to be unwanted, if a threshold amount of potentially unwanted data received from such source was determined to be unwanted, if all potentially unwanted data received from such source was determined to be unwanted, etc.

As yet another option, the data may be automatically identified as unwanted if it is determined that the data was received by a predefined threshold number of different sources. Such predefined threshold number may be user-configured, in one embodiment. For example, if the data was independently received (e.g. different copies of the data

were received) by the predefined threshold number of different sources, the data may be automatically identified as unwanted.

As still yet another option, the data may be automatically identified as unwanted if it is determined that a weight assigned to the source from which the data was received meets a predefined threshold weight. The predefined threshold weight may be user-configured, in one embodiment. Additionally, the weight assigned to the source may be based on any desired aspect of the source, such as a type of the source, an amount of potentially unwanted data previously received from the source that was determined to be unwanted, etc. As another option, the data may be automatically identified as unwanted if it is determined that an aggregate weight calculated from weights of each source from which the data was received meets the predefined threshold weight. Of course, however, the data may be automatically identified as unwanted in any desired manner.

In one embodiment, the data may be automatically identified as unwanted based on a probability that the data is actually unwanted. For example, if the source of the data includes a predetermined type of source, is associated with previously received data determined to be unwanted, etc., the probability that the data is unwanted may be determined to meet a threshold probability. In this way, prior to determining whether the data is unwanted via an analysis of the data, the data may optionally be automatically identified as unwanted.

Still yet, as shown in operation **306**, the data is stored for use in detecting unwanted data. With respect to the present description, the data may be stored in any desired type of data structure capable of allowing the data to be used in detecting unwanted data. In various embodiments, the data may be stored in a database, a list of known unwanted data (e.g. a blacklist), etc.

As an option, storing the data may include storing a hash of the data. As another option, a plurality of different types of hashes of the data may be stored. The hash may be computed utilizing message-digest algorithm 5 (MD5), secure hash algorithm-1 (SHA-1), secure hash algorithm-256 (SHA-256), etc.

Further, in one embodiment, an indication that the data is unwanted may be stored in association with the data. Such indication may include any type of identifier, for example. In another embodiment, an indication that the data is potentially unwanted data automatically determined to be unwanted data may be stored in association with the data.

Further still, the stored data may be used for detecting unwanted data by being identifiable as known unwanted data. As an option, other received data determined to be potentially unwanted may be compared with the stored data for determining whether such other received data is unwanted. For example, if the other received data matches the stored data, the other received data may be determined to be unwanted. As another example, if a hash of the other received data matches a hash of the stored data, the other received data may be determined to be unwanted. Thus, the stored data may optionally be used by the device (e.g. server) on which such data is stored for detecting unwanted data.

As another option, the stored data may be utilized by any other device (e.g. client device, etc.) for detecting unwanted data. Just by way of example, a remote client device may detect other potentially unwanted data (e.g. utilizing a security system, etc.), may calculate a hash of such potentially unwanted data, and may remotely query a database storing the stored data. If the query returns the stored data, the other device may determine that the other potentially unwanted data is unwanted. Of course, it should be noted that the stored data may be used in detecting unwanted data in any desired manner.

To this end, data determined to be potentially unwanted may be automatically identified as unwanted, prior to determining whether the data includes unwanted data via an analysis of such data. Moreover, storing the data automatically determined to be unwanted for use in detecting unwanted data may allow the data to be used in detecting unwanted data upon the storage of the data. Thus any delay in using the data for detecting unwanted data may be prevented, where such delay results from a delay in determining whether the data is actually unwanted (e.g. via an the analysis of such data), from a wait time resultant from a queue of stored data waiting to be processed for determining whether any of such data is actually unwanted, from a delay in providing an update of known unwanted data and/or known wanted data to client devices detecting the potentially unwanted data, from a delay in installing such update by the client devices, etc.

As an option, once the data is stored for use in detecting unwanted data, a subsequent analysis of the data may be performed for determining whether the data actually includes unwanted data. The subsequent analysis may be performed at any desired time, as the stored data may already be capable of being used to detect unwanted data. Just by way of example, the stored data may be identified by identifying data stored with an indication that the data includes potentially unwanted data automatically identified as unwanted.

In addition, the stored data may be analyzed, in response to identification thereof, and it may be determined whether the data is unwanted based on the analysis. Accordingly, if the data is determined to be unwanted, a list of known unwanted data may be updated. However, if it is determined that the data is wanted, a list of known wanted data may be updated. Such updated list of known unwanted data or known wanted data may further be provided to the source from which the data determined to be potentially unwanted was received (in operation **302**) and/or to any other desired device for local use in detecting unwanted data.

More illustrative information will now be set forth regarding various optional architectures and features with which the foregoing technique may or may not be implemented, per the desires of the user. It should be strongly noted that the following information is set forth for illustrative purposes and should not be construed as limiting in any manner. Any of the following features may be optionally incorporated with or without the exclusion of other features described.

FIG. **4** shows a system **400** for automatically identifying potentially unwanted (e.g. potentially malicious) data as unwanted (e.g. malicious), in accordance with another embodiment. As an option, the system **400** may be implemented in the context of the architecture and environment of FIGS. **1-3**. Of course, however, the system **400** may be implemented in any desired environment. It should also be noted that the aforementioned definitions may apply during the present description.

As shown, a server **404** is in communication with clients **402**A-**402**B. In one embodiment, the clients **402**A-**402**B may include any client capable of detecting potentially malicious data **406**A-**406**B that may be in communication with the server **404**. For example, the clients **402**A-**402**B may include one or more of the clients illustrated in FIG. **1**. Additionally, in another embodiment, the server **404** may

include any server capable of automatically identifying the potentially malicious data **406**A-**406**B as malicious and storing such data for use in detecting malicious data. For example, the server **404** may include the server illustrated in FIG. **1**.

Additionally, each of the clients **402**A-**402**B includes a security system **408**A-**408**B. In the context of the current embodiment, the security systems **408**A-**408**B may include any system utilized by the clients **402**A-**402**B to detect malicious data. For example, the security systems **408**A-**408**B may include a firewall, an anti-virus system, an anti-spyware system, etc.

In one embodiment, the security systems **408**A-**408**B may be constantly running on the clients **402**A-**402**B. In another embodiment, the security systems **408**A-**408**B may periodically run on the clients **402**A-**402**B. Of course, however, the security systems **408**A-**408**B may interact with the clients **402**A-**402**B in any manner.

To this end, each security system **408**A-**408**B may identify data **406**A-**406**B on an associated client **402**A-**402**B as potentially malicious. While the present embodiment is described below with respect to only one of the clients **402**A-**402**B, it should be noted that the clients **402**A-**402**B may operate in a similar manner. Thus, the present system **400** may be implemented with respect to either or both of the clients **402**A-**402**B.

In one embodiment, the security system **408**A of the client **402**A may identify the potentially malicious data **406**A by monitoring the client **402**A for malicious data. Further, the security system **408**A may determine that data **406**A on such client **402**A is potentially malicious in response to a determination that the data **406**A does not match known malicious data and does not match known clean (e.g. non-malicious) data. Such known malicious data and known clean data may be stored in a database on the client **402**A, for example.

In response to the identification of the potentially malicious data **406**A, the security system **408**A may send the potentially malicious data **406**A to the server **404**. In one embodiment, the potentially malicious data **406**A may be sent to the server **404** for determining whether the potentially malicious data **406**A is actually malicious. For example, the potentially malicious data **406**A may be sent to the server **404** for analyzing the potentially malicious data **406**A determine whether such is malicious.

Based on receipt of the potentially malicious data **406**A, the server **404** automatically identifies the potentially malicious data **406**A as malicious. For example, the server **404** may identify the potentially malicious data **406**A as malicious without necessarily analyzing the potentially malicious data **406**A. In one exemplary embodiment, the server **404** may identify the potentially malicious data **406**A as malicious based on an identification of the client **402**A from which the potentially malicious data **406**A was received as a previous source of malicious data.

Further, the server **404** stores the data automatically identified as malicious (or a hash thereof) in a list of known malicious data **410** located on the server **404**. In this way, the data may be stored in the list of known malicious data **410** for use in detecting malicious data. As an option, an identifier indicating that the data was potentially malicious data automatically identified as malicious may be stored in association with the data. Thus, the list of known malicious data **410** may optionally include data with an identifier indicating that the data was potentially malicious data automatically identified as malicious and data with an identifier indicating that the data is malicious (e.g. as determined based on an analysis of the data, etc.).

Still yet, once the server **404** is able to analyze the stored data automatically identified as malicious (e.g. in response to resources being available for such analysis, etc.), the server **404** may perform the analysis on the stored data. If the server **404** determines that the stored data is malicious, based on the analysis, the server **404** may create an updated data (DAT) file **414** (or update an existing DAT file) to include such data as known malicious data. As an option, the server **404** may also change the identifier stored with the data in the list of known malicious data **410** to indicate that the data is malicious (e.g. as determined based on an analysis of the data, as determined by other sources that periodically distribute updates to the list of known malicious data **410**, etc.).

If, however, the server **404** determines that the stored data is not malicious, based on the analysis, the server **404** may create the updated DAT file **414** (or update any existing DAT file) to include such data as known clean (e.g. non-malicious) data. Optionally, the server **404** may also remove the data from the list of known malicious data **410** and may store such data in a list of known clean data **412** (e.g. a list of data predetermined to be clean, etc.). As another option, the list of known clean data **412** may also be populated with data from software vendors (e.g. operating system vendors), data determined to be clean based on an analysis of such data by the server **404**, data determined to be clean based on a manual analysis (e.g. by human researchers) of such data, data from publicly available databases including known clean data (e.g. National Institute of Standards and Technology database, National Software Reference Library database, etc.), etc.

Furthermore, the server **404** may transmit the DAT **414** (e.g. as an update, etc.), which includes the data identified as malicious or clean, to the clients **402**A-**402**B. In this way, a list of known malicious data or a list of known clean data located on the clients **402**A-**402**B (not shown) may be updated for use in subsequent detections of malicious data.

Just by way of example, after storing the data in the list of known malicious data **410**, other data may be identified by a security system **408**A-**408**B of at least one of the clients **402**A-**402**B as potentially malicious. Based on the identification of the other potentially malicious data, the security system **408**A-**408**B may calculate a hash of the other potentially malicious data. In addition, the security system **408**A-**408**B may remotely query the server **404** for the hash [e.g. via a direct connection between the client **402**A-B and the server **404**, via a domain name server (DNS) cloud, etc.]. Of course, while the query is described herein as including the hash of the other potentially malicious data, it should be noted that the query may include the other potentially malicious data itself and/or any other information capable of being used to identify the other potentially malicious data.

The server **404** may subsequently receive the query, and may compare the hash received via the query with the list of known malicious data **410** and the list of known clean data **412**. If the server **404** determines that the received hash matches a hash in the list of known malicious data **410**, the server **404** may identify the other potentially malicious data associated with the hash as malicious. If, however, the server **404** determines that the received hash matches a hash in the list of known clean data **412**, the server **404** may identify the other potentially malicious data associated with the hash as clean. Further, a result to the query identifying the other potentially malicious data as malicious or clean may be sent to the client **402**A-**402**B from which the query was received.

It should be further noted that if the server **404** determines that the received hash does not match hashes include in either of the list of known malicious data **410** or the list of known clean data **412**, the server **404** may automatically determine that the other potentially malicious data associated with the hash is malicious, and may store a hash of the potentially malicious data in the list of known malicious data **410**, as described above.

FIG. **5** shows a method **500** for storing a hash of data with an indication of whether the data is potentially malicious or potentially clean, in accordance with yet another embodiment. As an option, the method **500** may be carried out in the context of the architecture and environment of FIGS. **1-4**. For example, the method **500** may be carried out using the server **404** of FIG. **4**. Of course, however, the method **500** may be carried out in any desired environment. It should also be noted that the aforementioned definitions may apply during the present description.

As shown in operation **502**, data is received. With respect to the present embodiment, the data may be received from a client device that determined that the data is potentially malicious. For example, the data may be received by the client device in response to a determination by the client device that it is unknown whether the data is malicious or clean.

Additionally, it is determined whether the data is known to be malicious or clean, as shown in decision **504**. For example, it may be determined whether the data has been predetermined to be malicious or clean. In one embodiment, the data may be compared with a list of known malicious data. For example, if the data matches data included in the list of known malicious data, the data may be determined to be known to be malicious.

In another embodiment, the data may be compared with a list of known clean data. Thus, if the data matches data included in the list of known clean data, the data may be determined to be known to be clean. If it is determined that the data is known to be malicious or clean, the method **500** terminates. As an option, an indication of whether the data is malicious or clean may be sent to the source from which the data was received (e.g. based on the determination), prior to the method **500** terminating.

If, however, it is determined that the data is not known to be malicious or clean, it is determined whether the data may be automatically identified as malicious. Note decision **506**. For example, determining whether the data may be automatically identified as malicious may include determining whether the data may be identified as malicious, at least temporarily, without performing an analysis on such data for determining whether the data is in fact malicious. In one embodiment, the data may be automatically identified as malicious based on a source of the data. Of course, however, the data may be automatically identified as malicious based on any desired aspect associated with the data that does not necessarily require an analysis of the data itself (e.g. an analysis of content of the data, etc.).

If it is determined that the data may not be automatically identified as malicious, the method **500** terminates. As an option, the client device from which the data was received may wait for the analysis to be performed on the data before such client device may receive an indication of whether the data is malicious. As another option, the client device may be notified that such analysis is required before any indication will be received by the client device.

If, however, it is determined that the data may be automatically identified as malicious, the data is hashed. Note operation **508**. Furthermore, the hash is stored in a database with an indication that the data is potentially malicious data automatically identified as malicious, as shown in operation **510**. To this end, the hash of the data may be stored such that the hash may be used for detecting unwanted data. As an option, an indication that the data has been automatically identified as malicious may be sent to the client device from which the data was received.

FIG. **6** shows a method **600** for querying a database of hashes for identifying potentially malicious data as malicious, in accordance with still yet another embodiment. As an option, the method **600** may be carried out in the context of the architecture and environment of FIGS. **1-5**. For example, the method **600** may be carried out using one of the clients **402**A-**402**B of FIG. **4**. Of course, however, the method **600** may be carried out in any desired environment. Again, it should be noted that the aforementioned definitions may apply during the present description.

As shown in decision **602**, it is determined whether potentially malicious data is detected. The data may be determined to be potentially malicious if it is determined that it is unknown whether the data is malicious or clean. For example, if the data does not match known malicious data or known clean data (e.g. stored on the device on which the data is located), the data may be determined to be malicious.

If it is determined that potentially malicious data is not detected, the method **600** continues to wait for potentially malicious data to be detected. If, however, it is determined that potentially malicious data is detected, a hash of the potentially malicious data is calculated. Note operation **604**.

Additionally, a server database of hashes of malicious data is queried for the calculated hash, as shown in operation **606**. Thus, the query may include a remote query. In one embodiment, the server database of hashes of malicious data may include any database storing hashes of known malicious data. For example, the server database of hashes of malicious data may include the list of known malicious data **410** of FIG. **4**.

Furthermore, as shown indecision **608**, it is determined whether a result of the query indicates that the calculated hash is found in the server database of hashes of malicious data. If, it is determined that the result of the query indicates that the calculated hash is not found in the server database of hashes of malicious data, the potentially malicious data detected in operation **602** is identified as undetermined to be malicious. Note operation **610**. As an option, the server storing the server database of hashes of malicious data may also identify the potentially malicious data as undetermined to be malicious if the result of the query indicates that the calculated hash is not found in the server database of hashes of malicious data.

Moreover, in response to a determination by the server that the potentially malicious data is undetermined to be malicious, the server may optionally automatically identify the potentially malicious data as malicious (e.g. as described above with respect to the method **500** of FIG. **5**). If it is determined that the result of the query indicates that the calculated hash is found in the server database of hashes of malicious data, the potentially malicious data detected in operation **602** is identified as malicious. Note operation **612**.

While various embodiments have been described above, it should be understood that they have been presented by way of example only, and not limitation. Thus, the breadth and scope of a preferred embodiment should not be limited by any of the above-described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents.

What is claimed is:

1. A computer program product *including computer code* embodied on a non-transitory computer readable medium, *and, when executed by at least one processor, the computer code causes the at least one processor to perform operations* comprising:

[computer code for**]** receiving, *from at least one source,* data by a server, wherein the data is not known to be wanted and wherein the data is not known to be unwanted;

[computer code for**]** assigning a weight to each *of the at least one* source from which the data was received;

[computer code for**]** calculating an aggregate weight from the weights assigned to each *of the at least one* source from which the data was received; **[**and

computer code for**]** automatically identifying by the server that the data is unwanted **[**if**]** *based on a determination that* the aggregate weight meets a predetermined threshold weight; and

[computer code for**]** storing the data by the server for use in detecting unwanted data, *wherein storing the data includes storing a hash of the data, in response to the identifying*.

2. The computer program product of claim **1**, wherein the data is not known to be malicious and wherein the data is not known to be non-malicious.

3. The computer program product of claim **1**, wherein the data does not match known wanted data and does not match known unwanted data.

4. The computer program product of claim **1**, wherein the data is automatically received based on monitoring of at least one of an electronic messaging application, file transfer protocol (FTP), and a web site.

5. The computer program product of claim **1**, wherein the data is automatically identified as unwanted based on *the* at least one source from which the data is received.

6. The computer program product of claim **5**, wherein the *at least one* source includes a security vendor.

7. The computer program product of claim **5**, wherein the *at least one* source includes a honeypot.

8. The computer program product of claim **5**, wherein the data is automatically identified as unwanted if it is determined that other data previously received from the at least one source **[**includes**]** *included* known unwanted data.

9. The computer program product of claim **5**, wherein the data is automatically identified as unwanted if it is determined that the data was received by a predefined threshold number of different sources.

10. The computer program product of claim **5**, wherein the data is automatically identified as unwanted if it is determined that a weight assigned to the at least one source meets a predefined threshold weight.

[11. The computer program product of claim **1**, wherein storing the data includes storing a hash of the data.**]**

12. The computer program product of claim **1**, **[**further comprising computer code for**]** *wherein the computer code causes the at least one processor to perform further operations comprising*

storing an indication with the data that the data includes potentially unwanted data automatically identified as unwanted.

13. The computer program product of claim **12**, **[**further comprising computer code for**]** *wherein the computer code causes the at least one processor to perform further operations comprising*

identifying the data stored with the indication that the data includes potentially unwanted data automatically identified as unwanted,

analyzing the data, and

determining whether the data is unwanted based on the analysis.

14. The computer program product of claim **13**, **[**further comprising computer code for**]** *wherein the computer code causes the at least one processor to perform further operations comprising*

updating one of a list of known unwanted data and a list of known wanted data based on the determination *whether the data is unwanted*.

15. The computer program product of claim **1**, wherein the stored data is utilized for detecting the unwanted data by identifying other received data determined to be potentially unwanted as unwanted if the other received data matches the stored data.

16. A method, comprising:

receiving, *from at least one source,* data by a computer processor, wherein the data is not known to be wanted and wherein the data is not known to be unwanted;

assigning a weight to each *of the at least one* source from which the data was received;

calculating an aggregate weight from the weights assigned to each *of the at least one* source from which the data was received;

identifying automatically that the data is unwanted **[**if**]** *based on a determination that* the aggregate weight meets a predetermined threshold weight; and

storing the data for use in detecting unwanted data, *wherein storing the data includes storing a hash of the data, in response to the identifying*.

17. A system, comprising:

a computer processor*; and*

[for**]** *logic that is executable by the computer processor and, when executed, causes the computer processor to perform operations including* receiving data *from at least one source*, wherein the data is not known to be wanted and wherein the data is not known to be unwanted, assigning a weight to each *of the at least one* source from which the data was received, calculating an aggregate weight from the weights assigned to each *of the at least one* source from which the data was received, identifying automatically that the data is unwanted **[**if**]** *based on a determination that* the aggregate weight meets a predetermined threshold weight, and storing the data for use in detecting unwanted data, *wherein storing the data includes storing a hash of the data, in response to the identifying*.

18. The method of claim **16**, further comprising:

identifying the data as unwanted by the computer processor by analyzing the data.

19. A method, comprising:

receiving a first data by a client computer;

analyzing the first data by the client computer, and determining that the first data is not known to be wanted and wherein the first data is not known to be unwanted;

sending the first data to a server computer;

receiving, by the server computer, the first data from at least one source including the client computer;

assigning a weight to each *of the at least one* source from which the *first* data was received *by the server computer*;

calculating an aggregate weight from the weights assigned to each *of the at least one* source from which the *first* data was received *by the server computer*;

identifying automatically that the *first* data is unwanted **[**if**]** *based on a determination that* the aggregate weight meets a predetermined threshold weight; and

storing the first data by the server computer, *wherein storing the data includes storing a hash of the data, in response to the identifying.*

20. The method of claim **19**, further comprising:

analyzing the stored first data by the server computer responsive to analysis resources being available; **[**and**]**

identifying the stored first data as wanted or unwanted responsive to the **[**act of**]** analyzing the stored first data by the server computer;

updating a datastore responsive to the **[**act of**]** identifying the stored first data as wanted or unwanted; *and*

distributing the updated datastore to the client computer.

21. The method of claim **19**, *further comprising:*

receiving a second data from the client computer, wherein the second data comprises an identification of the client computer as a previous source of unwanted data.

22. The method of claim **19**, further comprising:

using the stored first data automatically identified as unwanted for determining that a third data is unwanted.

23. The computer program product of claim **1**, wherein the **[**computer code for**]** automatically identifying by the server *that* the data **[**as**]** *is* unwanted **[**without analyzing the data**]** comprises:

[computer code for**]** automatically identifying by the server the data as unwanted **[**responsive to**]** *without analyzing the data, based on* a second data**[**, without analyzing the first data**]**.

24. The method of claim **16**, wherein the **[**act of**]** identifying the data automatically as unwanted **[**by the computer processor without analyzing the data**]** comprises:

identifying the data automatically as unwanted by the computer **[**process**]** *processor* without analyzing the data, **[**responsive to**]** *based on* a second data.

25. The method of claim **24**, wherein the second data comprises a source from which the data is received.

26. The system of claim 17, wherein the data is not known to be malicious and wherein the data is not known to be non-malicious.

27. The computer program product of claim 1, wherein the computer code further causes the at least one processor to perform further operations comprising

determining that other received data is unwanted, if a hash of the other received data matches the hash of the data.

28. The computer program product of claim 1, wherein the computer code further causes the at least one processor to perform further operations comprising

performing an analysis of the data for determining whether the data is actually the unwanted data, once the data is stored.

29. The system of claim 17, wherein the data does not match known wanted data and does not match known unwanted data.

30. The system of claim 17, wherein the data is automatically received based on monitoring of at least one of an electronic messaging application, file transfer protocol (FTP), and a web site.

31. The system of claim 17, wherein the data is automatically identified as unwanted based on the at least one source from which the data is received.

32. The system of claim 31, wherein the at least one source includes a security vendor.

33. The system of claim 31, wherein the at least one source includes a honeypot.

34. The system of claim 31, wherein the data is automatically identified as unwanted if it is determined that other data previously received from the at least one source included known unwanted data.

35. The system of claim 31, wherein the data is automatically identified as unwanted if it is determined that the data was received by a predefined threshold number of different sources.

36. The system of claim 31, wherein the data is automatically identified as unwanted if it is determined that a weight assigned to the at least one source meets a predefined threshold weight.

37. The system of claim 17, wherein the operations further include storing an indication with the data that the data includes potentially unwanted data automatically identified as unwanted.

38. The system of claim 37, wherein the operations further include identifying the data stored with the indication that the data includes potentially unwanted data automatically identified as unwanted, analyzing the data, and determining whether the data is unwanted based on the analysis.

39. The system of claim 38, wherein the operations further include updating one of a list of known unwanted data and a list of known wanted data based on the determination whether the data is unwanted.

40. The system of claim 17, wherein the stored data is further utilized for detecting the unwanted data by identifying other received data determined to be potentially unwanted as unwanted if the other received data matches the stored data.

41. The system of claim 17, wherein the operations further include:

analyzing the stored data responsive to analysis resources being available;

identifying the stored data as wanted or unwanted responsive to the analyzing the stored data;

updating a datastore responsive to the identifying the stored data as wanted or unwanted; and

distributing the updated datastore to a client computer.

42. The system of claim 17, wherein the operations further include receiving a second data from a client computer, and the second data comprises an identification of the client computer as a previous source of unwanted data.

43. The system of claim 17, wherein the operations further include using the stored data automatically identified as unwanted for determining that a third data is unwanted.

44. The system of claim 17, wherein the operations further include:

identifying the data automatically as unwanted without analyzing the data, based on a second data.

45. The system of claim 44, wherein the second data comprises a source from which the data is received.

46. The system of claim 17, wherein the operations further include identifying the data as unwanted by analyzing the data.

* * * * *