US00RE47628E

(19) United States
(12) Reissued Patent
Hopkins

(10) Patent Number: US RE47,628 E
(45) Date of Reissued Patent: Oct. 1, 2019

(54) **SYSTEM FOR IDENTIFYING THE PRESENCE OF PEER-TO-PEER NETWORK SOFTWARE APPLICATIONS**

(75) Inventor: **Samuel P. Hopkins**, Zelienople, PA (US)

(73) Assignee: **Kroll Information Assurance, LLC**, New York, NY (US)

(21) Appl. No.: **13/445,343**

(22) Filed: **Apr. 12, 2012**

**Related U.S. Patent Documents**

Reissue of:
(64) Patent No.: **7,697,520**
Issued: **Apr. 13, 2010**
Appl. No.: **11/599,828**
Filed: **Nov. 15, 2006**

U.S. Applications:
(63) Continuation-in-part of application No. 11/103,818, filed on Apr. 12, 2005, now Pat. No. 9,178,940.

(60) Provisional application No. 60/736,794, filed on Nov. 15, 2005.

(51) **Int. Cl.**

| | |
|---|---|
| ***H04L 12/28*** | (2006.01) |
| ***G06F 21/57*** | (2013.01) |
| ***H04L 29/08*** | (2006.01) |
| ***H04L 29/06*** | (2006.01) |
| ***H04L 12/24*** | (2006.01) |
| ***H04L 12/26*** | (2006.01) |

(52) **U.S. Cl.**
CPC ..... ***G06F 21/57*** (2013.01); ***H04L 41/046*** (2013.01); ***H04L 43/12*** (2013.01); ***H04L 63/1416*** (2013.01); ***H04L 67/104*** (2013.01); ***H04L 67/1061*** (2013.01)

(58) **Field of Classification Search**
CPC . H04L 12/28; H04L 67/1061; H04L 63/1416; H04L 67/104; H04L 41/046; H04L 43/12; G06F 15/177; G06F 15/173; G06F 15/16; G06F 21/57
USPC ..... 709/223, 220, 224, 225; 370/389, 346; 726/26
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

| | | | |
|---|---|---|---|
| 6,044,471 | A | 3/2000 | Colvin |
| 6,061,794 | A | 5/2000 | Angelo et al. |
| 6,151,643 | A | 11/2000 | Cheng et al. |
| 6,289,446 | B1 | 10/2001 | Schreiber et al. |

(Continued)

FOREIGN PATENT DOCUMENTS

| | | |
|---|---|---|
| JP | 2004-046419 | 2/2004 |
| JP | 2004-343186 | 12/2004 |

(Continued)

OTHER PUBLICATIONS

Zupeng, Li et al., "Research of Peer-to-Peer Network Architecture," Proceedings of ICCT2003, pp. 312-315.

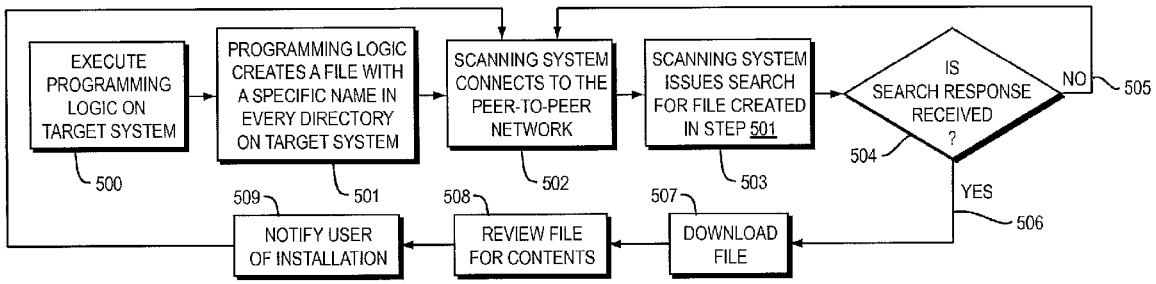
(Continued)

*Primary Examiner* — Mark Sager
(74) *Attorney, Agent, or Firm* — Arnold & Porter Kaye Scholer

(57) **ABSTRACT**

A system and method for detecting peer-to-peer network software operating on a target computer. A target file is created, and placed in one or more folders on the target computer. A search is issued on a Peer-to-Peer network for the target file. Peer-to-peer software is detected to be operating on the target computer in accordance with results of the search.

**19 Claims, 4 Drawing Sheets**



(56) **References Cited**

U.S. PATENT DOCUMENTS

| | | | | |
|---|---|---|---|---|
| 6,574,729 | B1 | 6/2003 | Fink et al. | |
| 6,668,289 | B2 | 12/2003 | Cheng et al. | |
| 6,792,545 | B2 | 9/2004 | McCreight et al. | |
| 6,918,113 | B2 | 7/2005 | Patel et al. | |
| 6,983,326 | B1 | 1/2006 | Vigue et al. | |
| 7,042,852 | B2 | 5/2006 | Hrastar | |
| 7,046,995 | B2 | 5/2006 | Rygaard | |
| 7,051,098 | B2 | 5/2006 | Masters et al. | |
| 7,086,089 | B2 | 8/2006 | Hrastar et al. | |
| 7,089,301 | B1 | 8/2006 | Labio et al. | |
| 7,096,503 | B1 | 8/2006 | Magdych et al. | |
| 7,213,047 | B2 | 5/2007 | Yeager et al. | |
| 7,263,070 | B1 | 8/2007 | Delker et al. | |
| 7,277,404 | B2 * | 10/2007 | Tanzella et al. | 370/310 |
| 7,281,268 | B2 | 10/2007 | Hollander et al. | |
| 7,296,288 | B1 | 11/2007 | Hill et al. | |
| 7,318,092 | B2 * | 1/2008 | Sutler | 709/223 |
| 7,325,251 | B1 | 1/2008 | Szor | |
| 7,543,052 | B1 | 6/2009 | Cesa Klein | |
| 7,574,523 | B2 | 8/2009 | Traversat et al. | |
| 7,664,048 | B1 | 2/2010 | Yung et al. | |
| 7,840,663 | B1 * | 11/2010 | Hinchliffe et al. | 709/224 |
| 7,903,822 | B1 * | 3/2011 | Hair et al. | 380/282 |
| 8,667,103 | B2 | 3/2014 | Adar et al. | |
| 2001/0003191 | A1 | 6/2001 | Kovacs et al. | |
| 2002/0044549 | A1 | 4/2002 | Johansson et al. | |
| 2002/0069098 | A1 | 6/2002 | Schmidt | |
| 2002/0073204 | A1 | 6/2002 | Dutta et al. | |
| 2002/0087885 | A1 * | 7/2002 | Peled et al. | 713/201 |
| 2002/0184310 | A1 | 12/2002 | Traversat et al. | |
| 2003/0041141 | A1 * | 2/2003 | Abdelaziz et al. | 709/223 |
| 2003/0055892 | A1 | 3/2003 | Huitema et al. | |
| 2003/0093466 | A1 * | 5/2003 | Jarman et al. | 709/203 |
| 2003/0095660 | A1 * | 5/2003 | Lee et al. | 380/231 |
| 2003/0112974 | A1 * | 6/2003 | Levy | 380/210 |
| 2003/0195852 | A1 * | 10/2003 | Campbell et al. | 705/51 |
| 2003/0208621 | A1 | 11/2003 | Bowman | |
| 2004/0003266 | A1 | 1/2004 | Moshir et al. | |
| 2004/0030651 | A1 | 2/2004 | Kim et al. | |
| 2004/0039921 | A1 | 2/2004 | Chuang | |
| 2004/0044996 | A1 | 3/2004 | Atallah | |
| 2004/0107347 | A1 * | 6/2004 | Ogino | 713/176 |
| 2004/0122958 | A1 | 6/2004 | Wardrop | |
| 2004/0133640 | A1 | 7/2004 | Yeager et al. | |
| 2004/0148434 | A1 | 7/2004 | Matsubara et al. | |
| 2004/0153658 | A1 | 8/2004 | Gunyakti et al. | |
| 2004/0162871 | A1 * | 8/2004 | Pabla et al. | 709/201 |
| 2005/0044483 | A1 * | 2/2005 | Maze et al. | 715/501.1 |
| 2005/0071485 | A1 | 3/2005 | Ramagopal | |
| 2005/0091397 | A1 | 4/2005 | Roberts et al. | |
| 2005/0102383 | A1 * | 5/2005 | Sutler | 709/223 |
| 2005/0105476 | A1 * | 5/2005 | Gotesdyner et al. | 370/254 |
| 2005/0108557 | A1 * | 5/2005 | Kayo et al. | 713/189 |
| 2005/0177752 | A1 | 8/2005 | Hollander et al. | |
| 2005/0193216 | A1 | 9/2005 | Gurda et al. | |
| 2005/0198535 | A1 | 9/2005 | Basche et al. | |
| 2005/0251486 | A1 | 11/2005 | Nair | |
| 2006/0059560 | A1 * | 3/2006 | Montulli | 726/26 |
| 2006/0103732 | A1 | 5/2006 | Bateman | |
| 2006/0168304 | A1 | 7/2006 | Bauer et al. | |
| 2006/0200865 | A1 | 9/2006 | Leake et al. | |
| 2006/0242121 | A1 * | 10/2006 | DeVorchik et al. | 707/3 |
| 2006/0248062 | A1 | 11/2006 | Libes et al. | |

FOREIGN PATENT DOCUMENTS

| | | |
|---|---|---|
| JP | 2006-311048 | 11/2006 |
| JP | 2006-330783 | 12/2006 |
| WO | 199203001 | 2/1992 |
| WO | WO 2003/063023 A2 | 7/2003 |
| WO | WO 2004/047408 | 6/2004 |
| WO | WO 2004/086168 | 10/2004 |
| WO | 2005045624 | 5/2005 |

OTHER PUBLICATIONS

Marmor, M.S., "Make the P2P Lead with Toadnode," www.webtechniques.com, Dec. 2000, pp. 44-49.

Ueda, K. et al., "Peer-to-Peer Network Topology Control within a Mobile Ad-hoc Network," 2003 IEEE, pp. 243-247.

Liu, J. et al., "Distributed Distance Measurement for Large-Scale Networks," Computer Networks 41 (2003) 177-192.

Siu Man Lui and Sai Ho Kowk, "Interoperability of Peer-to-Peer File Sharing," ACM SIGecom Exchanges, vol. 3, No. 3, Aug. 2002, pp. 25-33.

Brandon Wiley, Freenet, "Inoperability Through Gateways," Chapter 19, pp. 381-392.

Zhenyun Zhuang et al., "Hybrid Periodical Flooding in Unstructured Peer-to-Peer Networks," Proceedings of the 2003 International Conference on Parallel Proceeding.

Hessing, S., "Peer to Peer Messaging Protocol," Internet-Draft, Apr. 2002, pp. 1-57.

Oram, A., "Peer-to-Peer: Harnessing the Power of Distruptive Technologies," Ch. 19: Inoperability Through Gateways, Mar. 2001, p. 381-392.

Lindemann, C. et al., "A Distributed Search Service for Peer-to-Peer File Sharing in Mobile Applications," Peer-to-Peer Computing, 2002, Proceedings of the Second International Conference on Peer-to-Peer Computing, Sep. 5-7, 2002.

Findeli, M., "Peer-to-Peer (P2P) Networking," Online, Jul. 1, 2001, p. 1-21.

Scarlata, V. et al., "Responder Anonymity and Anonymous Peer-to-Peer File Sharing," Proceedings of the International Conference on Network Protocols, Nov. 11, 2001, p. 272-280.

Xiao, L. et al., "Mutual Anonymity Protocols for Hybrid Peer-to-Peer Systems," Proceedings of the 23rd International Conference on Distributed Computing Systems, May 19-22, 2003.

Richard, Matthew, "Intrusion Detection FAQ: Are there limitations of Intrusion Signatures?" Sans Institute, Apr. 5, 2001, https://www.sans.org/security-resources/idfaq/limitations.php., 5 pages.

Mennecke, Thomas, "P2P Users Face New Challenge—Update," Slyck News, Dec. 15, 2003, http://www.slyck.com/story345_P2P_Users_Face_New_Challenge_Update, 3 pages.

Notice of Reasons for Rejection, dated Jun. 24, 2013 for Japanese Patent Application No. 2011-239203, 5 pages.

Katsuo Yamada, "Latest P2P Mechanism and Corporate Defense Measures," "N + I Network Guide," Japan, Softbank Publishing Corp., Nov. 1, 2004, 2004. 11, pp. 084 to 093—Partial English explanation available (See Notice of Reasons for Rejection, dated Jun. 24, 2013 for Japanese Patent Application No. 2011-239203, 5 pages.).

TTS, "Latest P2P Mechanism and Corporate Defense Measures," "N + I Network Guide," Japan, Softbank Publishing Corp., Nov. 1, 2004, 2004. 11, pp. 071 to 078—Partial English explanation available (See Notice of Reasons for Rejection, dated Jun. 24, 2013 for Japanese Patent Application No. 2011-239203, 5 pages.).

TTS, "Latest P2P Mechanism and Corporate Defense Measures," "N +I Network Guide," Japan, Softbank Publishing Corp., Nov. 1, 2004, 2004. 11, pp. 094 to 097—Partial English explanation available (See Notice of Reasons for Rejection, dated Jun. 24, 2013 for Japanese Patent Application No. 2011-239203, 5 pages.).

Decision on Appeal—Reversed, dated Mar. 4, 2014 for U.S. Appl. No. 11/103,818, filed Apr. 12, 2005 for System and Method for Detecting Peer-To-Peer Network Software, 8 pages.

Supplementary Search Report dated Dec. 20, 2012 for European Patent Application No. 06 749 896.4 filed on Apr. 11, 2006 by Tiversa, Inc., 7 pages.

Youngfen Chen, et al. "Detecting randomly scanning worms based on heavy-tailed property", Networking, Sensing and Control, 2005, Proceedings 2005 IEEE Tucson, AZ Mar. 19-22, 2005, Piscataway, NJ, IEEE, Mar. 19, 2005, pp. 354-358.

Kotegawa et al., "Design for end host security management mechanism with mobile agents," a collection of papers of Computer Security Symposium 2004 (CSS2004) vol. II of II, Information Processing Society of Japan, Oct. 20, 2004, vol. 2004, No. 11, pp.

(56) **References Cited**

OTHER PUBLICATIONS

637-642—Partial English Translation available (See Office Action for Japanese Patent Application No. 2008-506642, dated Feb. 15, 2010, 8 pages.).
Office Action for Japanese Patent Application No. 2008-506642, dated Feb. 15, 2010, 8 pages.
Miyamoto, "Method of detecting/blocking unexpected communication," the Jun. 2004 issue of SoftwareDesign, Gijutsu-Hyohron CO., Ltd., Jun. 18, 2004, No. 230, pp. 33-36—Partial English Translation available (See Office Action for Japanese Patent Application No. 2008-506642, dated Feb. 15, 2010, 8 pages.).
Oda et al., "Part 3 Putting home server to practical use on the Internet with dynamic DNS," the Apr. 2003 issue of PC Japan, Softbank Publishing Co., Ltd., Apr. 1, 2003, vol. 8, No. 4, pp. 147-155—Partial English Translation available (See Office Action for Japanese Patent Application No. 2008-506642, dated Feb. 15, 2010, 8 pages.).
International Search Report and Written Opinion dated Sep. 26, 2007 for PCT/US06/44366 filed on Nov. 15, 2006 by Tiversa, Inc., 8 pages.
International Search Report and Written Opinion dated Sep. 24, 2007 for PCT/US06/13671 filed on Apr. 11, 2006 by Tiversa, Inc., 8 pages.
Spognardi, Angelo, Alessandro Lucarelli and Roberto Di Pietro. "A methodology for P2P file-sharing traffic detection." Hot Topics in Peer-to-Peer Systems, 2005. Hot-P2P 2005. Second International Workshop on IEEE, 2005.
Sen, Subhabrata, Oliver Spatscheck and Dongmei Wang/ "Accurate, scalable in-network identification on p2p traffic using application signatures." Proceedings of the 13th international conference on World Wide Web ACM 2004.
Supplementary European Search Report dated Jul. 7, 2016 for European Application No. EP 06 83 7687 by Tiversa IP, Inc. 8 pages.
Mockapetris et al., "Development of the Domain Name System," *ACM SigComm Computer Communication Review* (2001), Herewith.
Office Action dated Oct. 19, 2018 in Brazil Appln. PI06186130 (including translation), Herewith.
Steven Hessing, "Peer to Peer Messaging Protocol," Internet-Draft, 1-57 (2002).

* cited by examiner

FIG. 1
(PRIOR ART)

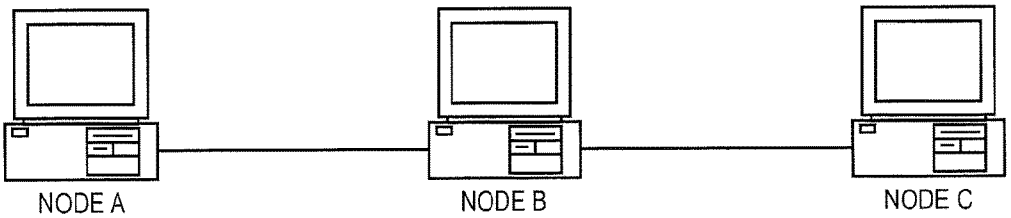



FIG. 2
(PRIOR ART)

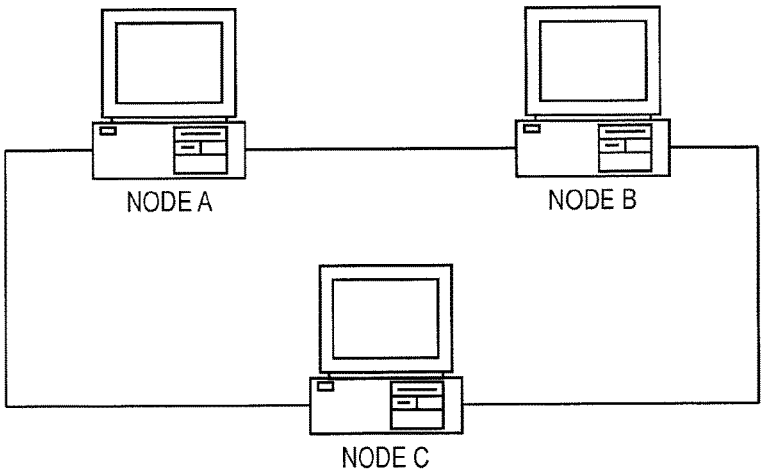



FIG. 3
(PRIOR ART)

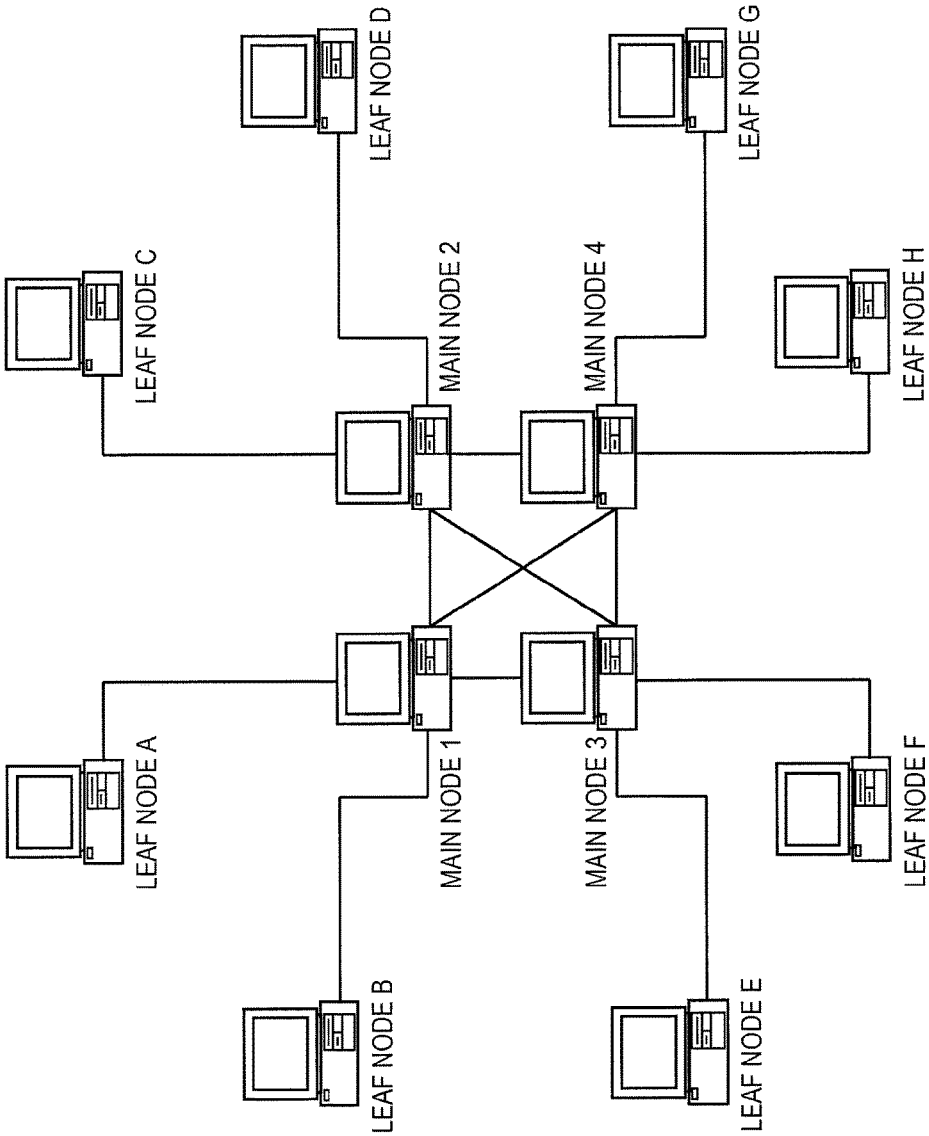

LEAF NODE A
LEAF NODE B
LEAF NODE C
LEAF NODE D
MAIN NODE 1
MAIN NODE 2
MAIN NODE 3
MAIN NODE 4
LEAF NODE E
LEAF NODE F
LEAF NODE G
LEAF NODE H

FIG. 4

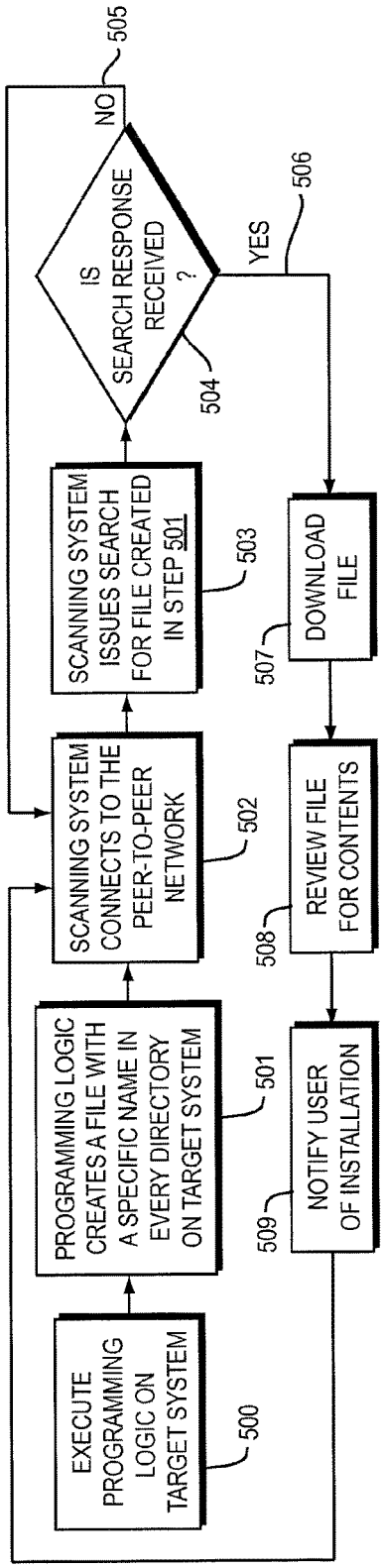

EXECUTE PROGRAMMING LOGIC ON TARGET SYSTEM
500
PROGRAMMING LOGIC CREATES A FILE WITH A SPECIFIC NAME IN EVERY DIRECTORY ON TARGET SYSTEM
501
SCANNING SYSTEM CONNECTS TO THE PEER-TO-PEER NETWORK
502
SCANNING SYSTEM ISSUES SEARCH FOR FILE CREATED IN STEP 501
503
IS SEARCH RESPONSE RECEIVED ?
504
NO
505
YES
506
DOWNLOAD FILE
507
REVIEW FILE FOR CONTENTS
508
NOTIFY USER OF INSTALLATION
509

FIG. 5

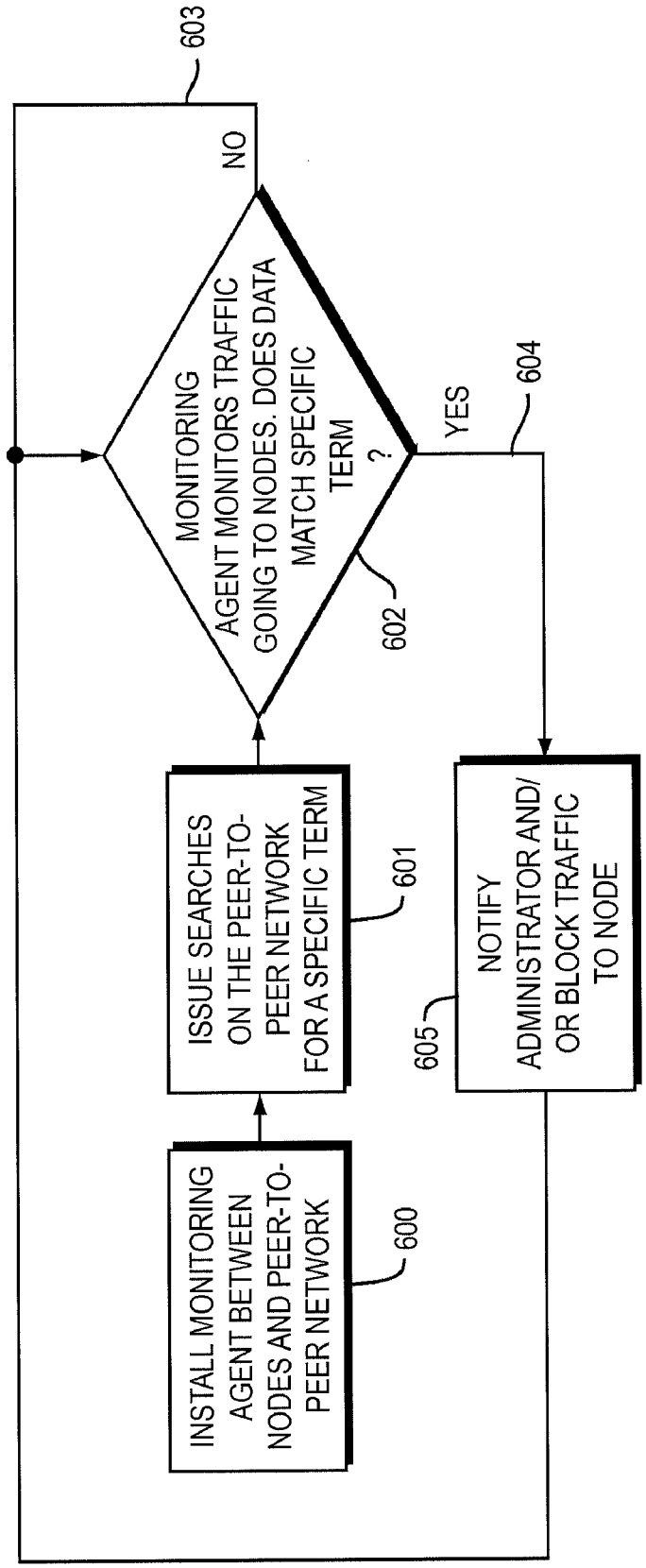

INSTALL MONITORING AGENT BETWEEN NODES AND PEER-TO-PEER NETWORK
600
ISSUE SEARCHES ON THE PEER-TO-PEER NETWORK FOR A SPECIFIC TERM
601
MONITORING AGENT MONITORS TRAFFIC GOING TO NODES. DOES DATA MATCH SPECIFIC TERM ?
602
NO
603
YES
604
NOTIFY ADMINISTRATOR AND/ OR BLOCK TRAFFIC TO NODE
605

FIG. 6

# SYSTEM FOR IDENTIFYING THE PRESENCE OF PEER-TO-PEER NETWORK SOFTWARE APPLICATIONS

**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 PATENT APPLICATIONS

The application claims priority to U.S. Provisional Appl. No. 60/736,794, filed Nov. 15, 2005, entitled "System for Identifying the Presence of Peer-to-Peer Network Software Applications," and is a continuation in part of U.S. patent application Ser. No. 11/103,818 filed Apr. 12, 2005 entitled System for Detecting Peer-to-Peer Network Software. Both of the above-referenced applications are incorporated by referenced herein in their entirety.

## FIELD OF THE INVENTION

The present invention provides a system for detecting whether or not a computer system is or could participate in a Peer-to-Peer network by searching for specific terms and detecting these terms, or the resulting files.

## BACKGROUND OF THE INVENTION

Peer-to-Peer networks comprise multiple nodes, each node typically consisting both of file server and client which can send and receive data or "Communication messages" to or from a node to which such is connected and other nodes on the network. Common Peer-to-Peer networks and software applications are Gnutella, FastTrack, Edonkey, Neo-Net, Kazaa, Limewire, Morpheus, Bear Share, Bit Torrent, Shareaza, Emule, and Freenet.

In a Peer-to-Peer network, each node is connected to other nodes over a communication medium such as the Internet either directly or through some type of proxy. For example, when a search request is issued such originating node sends a search request to all of the nodes to which it is connected. (See FIG. **1**) These nodes search their list of available files and if a match is found they send a response back with the location. However, a Peer-to-Peer proxy network typically consists of node A which is connected to a node B and node B is connected to a node C. (See FIG. **2**) Node A is not connected to node C such that if node A issues a search request it will be forwarded to node B and Node B will search its available files and if a match is found it will send a response back to node A. Node B will then forward node A's request to node C and Node C will search its available files and if a match is found it will send a response back to node B. Node B will then forward this response to node A. FIG. **3** discloses a nonproxy loop network wherein each node is directly connected to another.

Some Peer-to-Peer networks utilize a leaf node/main node proxy topology (see FIG. **4**) where some nodes are classified as main nodes and the remaining nodes are classified as leaf nodes. Leaf nodes can only connect to main nodes. Only main nodes can connect to other main nodes. When a leaf node issues a search request it sends the request to the main node to which it is connected. The main node then forwards the request to any other leaf nodes that are connected to it and also to any main nodes to which it is connected. These main nodes forward the request to any leaf nodes that are connected to them.



A Peer-to-Peer network is used to share files among its users. They are commonly used to share and acquire copyrighted music, movies, ebooks, and software but can be used to share and acquire almost any other type of file. To access a Peer-to-Peer network, a user installs a Peer-to-Peer network software application that is capable of connecting to and utilizing the Peer-to-Peer network, much the same way that a user installs a web browser, such as Internet Explorer, to access the World Wide Web.

Organizations are placed at legal risk by Peer-to-Peer network usage by their employees if an employee installs a Peer-to-Peer network software application onto their work PC and utilizes the Peer-to-Peer network to acquire copyrighted works. Peer-to-Peer network usage also consumes a lot of network bandwidth because the commonly transferred files are large software and movie files. This places bandwidth burdens on an organization's computer network. Even though it is normally a violation of corporate policy to have a Peer-to-Peer network software application installed, employees still install these applications.

When installing a Peer-to-Peer network software application, the user must select a folder on their computer system in which to store any downloaded files. For the purposes of clarification, a "folder" is used to organize files on a computer system, also known as a "directory." Any files placed into this folder are also made available to other users. This folder is often called the "Shared Folder". For instance, if user #**1** (on a first network node) places a file named "foofile" in their shared folder, user #**2** (on a second network node) would then be able to access and download the file. Depending on the Peer-to-Peer network software application used, the user can also select additional folders to make available to other users of the network.

For whatever reason, users sometimes select as their shared folder a folder that contains sensitive information or information they do not otherwise wish to share or they may later begin to place sensitive information or information they do not otherwise wish to share into their shared folder by mistake. Usually this action is done by mistake and unknowingly by the user but sometimes it is done by a malicious person or virus. Sometimes the Peer-to-Peer network software application has a software bug that permits the sharing of files and folders that the user never intended to be shared. Unintended (or malicious) sharing of information may be detrimental the user, the organization they work for, or even to national security. It would therefore be advantageous to be able to locate computers with Peer-to-Peer network software applications installed so that such applications can be assessed or removed.

There are hundreds if not thousands of different Peer-to-Peer network software applications with each having its own set of attributes. Current detection methods concentrate on 1) identifying the presence of each of these different Peer-to-Peer network software applications on a computer system or 2) the placement of a hardware/software based inline filter between the computer system and Internet to detect Peer-to-Peer network communications by looking for their protocols, monitoring for downloads, or increased bandwidth usage.

As Peer-to-Peer network software applications are created or current ones change, detecting the presence of a specific Peer-to-Peer network software application on a computer

system or monitoring for Peer-to-Peer network communication on the organization's network becomes increasing challenging.

The method of identifying the presence of Peer-to-Peer network software applications on a computer system entails creating a software "blueprint" of each Peer-to-Peer network software application and checking to see if this blueprint exists on a target computer system. Virus scanning software works in the same way, in that a blueprint of the virus is created and then checked against each file on a target computer system. Using a software blueprint to detect Peer-to-Peer network software applications is successful only if the Peer-to-Peer network software application is known and an accurate blueprint has been created. Each time a new Peer-to-Peer network software application is created a new blueprint must be created and there is an inherent lag in protection during the development of the software blueprint. Furthermore, when a Peer-to-Peer network software application is upgraded or changed because of new developments, a Peer-to-Peer network software application blueprint may no longer be valid. This leaves an organization exposed.

Inline filters detect Peer-to-Peer network usage by monitoring network communications on the organization's network and comparing the communications to known Peer-to-Peer network protocols. Using a protocol comparison method to detect Peer-to-Peer network software application only works if the Peer-to-Peer network software application's protocol is known. Each time a new Peer-to-Peer network software application is created the inline filter must be upgraded to look for the new protocol or data. Furthermore, when a Peer-to-Peer network software application is upgraded or changed because of new developments, the comparison filter that the inline filter uses may no longer be valid. Inline filters also do not work on Peer-to-Peer networks in which the communications between users is encrypted. This leaves an organization exposed.

## SUMMARY OF THE INVENTION

One aspect of the present invention is directed to a system and method for detecting peer-to-peer network software operating on a target computer. A target file is created, and placed in one or more folders on the target computer. A search is issued on a Peer-to-Peer network for the target file. Peer-to-peer software is detected to be operating on the target computer in accordance with results of the search.

In specific embodiments, the target file may be placed in a plurality of folders on the target computer, and optionally contains data that uniquely identifies the target computer. The data may be encrypted, and can include an IP address of the target computer, a name of the target computer, a name of a user of the target computer, and/or an email address of a user of the target computer. The data may be entered by a network administrator or operator responsible for monitoring the target computer. The method/system may be implemented at least in part using software that executes on the target computer, or alternatively using software that executes on a computer different from the target computer. A firewall, an intrusion detection system, a router, or an application, may be automatically notified upon detection of peer-to-peer software on the target computer.

In accordance with a further aspect, the present invention is directed to a system and method for detecting peer-to-peer network participation of a first node. Network data of the first node is monitored. A search is issued for a predetermined term on a peer-to-peer network while monitoring the network data. Peer-to-peer participation of the first node in the peer-to-peer network is detected if the monitoring identifies the predetermined term being transmitted to the first node.

In specific embodiments, an administrator responsible for monitoring the first node is notified if peer-to-peer participation is detected. Blocking of data access to the first node may be automatically implemented if peer-to-peer participation is detected. In addition, peer-to-peer software associated with the first node may be automatically or manually disabled if peer-to-peer participation is detected.

## BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a simplified schematic of a two node Peer-to-Peer network;

FIG. **2** is a simplified schematic of a Peer-to-Peer proxy network;

FIG. **3** is a simplified schematic view of a Peer-to-Peer, nonproxy, loop network;

FIG. **4** is a simplified schematic of a peer to peer leaf/main node network.

FIG. **5** is a simplified flowchart representation of one of embodiment of the present invention where a file is placed onto a target system and is then searched for via the peer-to-peer network; and

FIG. **6** is a simplified flowchart representation of another embodiment of the present invention where a monitoring agent is placed between a target system and a peer-to-peer network. A search is then initiated on the peer-to-peer network to see if the monitoring agent detects the search coming into the network.

## DESCRIPTION OF PRESENTLY PREFERRED EMBODIMENTS

As part of operating, nodes on a peer-to-peer network receive searches from the network for items that are being searched for by other users. If a first node receives a search and has a matching item, the first node will respond back to the searcher node. Referring to FIG. **6**, in one embodiment of the present invention, administrators install (**600**) a monitoring agent and configure it to detect certain terms heading inbound to a node or group of nodes that they wish to protect. The monitoring agent could be a device on its own, a piece of software, embedded in a router or firewall, or other network device which passes network data or has the potential to monitor network data such as a sniffer. The monitoring agent could have the data (being monitored) pass through it, or it could receive a copy of such data. The administrators then issues (**601**) searches on the peer-to-peer network and see (**602**) if these terms get sent to any protected nodes. If the term is not detected (**603**) by the monitoring agent as being sent to a node that is being protected, then the administrators continue to see (**602**) if these terms get sent to any protected nodes. If the term detected (**604**) by the monitoring agent as being sent to a node that is being protected, this would signal to the administrators that the node has peer-to-peer software. They would know this because the monitoring agent had detected the search inbound to a node so therefore the node must be part of the peer-to-peer network to receive the search. Upon detection that a node has peer-to-peer software, the monitoring agent is optionally configured to block (**605**) all transmissions to the node until the administrators removed the sofiware. The monitoring agent could also be configured to notify (**605**) the administrators upon detection that a node

has peer-to-peer software. The monitoring agent could be configured to do either of these functions (i.e., blocking/notification) automatically upon detection that a node has peer-to-peer software.

Referring to FIG. **5**, one embodiment of the present invention advantageously utilizes a software program to create a target file and places this target file in the folders of a target computer that is to be monitored for the purpose of detecting whether the target computer contains Peer-to-Peer network application software. The target file is preferably placed in as many folders as possible in the target computer because the "shared folder" on the target computer (to the extent one exists) is not known to the monitoring application. If the target computer has a Peer-to-Peer network software application installed, this target file will be available to be shared with other users of the Peer-to-Peer network, and an appropriate search of the Peer-to-Peer network for the target file will result in detection of the target file by a monitoring application. Once the target file or its data is detected on a Peer-to-Peer network, it can be assumed that the target computer is in some way participating in the Peer-to-Peer network and steps can be taken to remove the Peer-to-Peer network software application from the target computer. The advantages of this system compound when the monitoring application protects multiple target computers such as you would find on a corporate network. It can also be used to offer home users or consumers protection from inadvertent or malicious installation of a peer-to-peer client application.

In another embodiment, a software program is executed on the target computer. The software program creates a target file. The target file is placed into the folders of the target computer. For example, the target file is placed in as many folders as possible because the "shared folder" is not known. A search is initiated on a Peer-to-Peer network to check for the presence of the target file. If the target file is located (e.g., a node searching for the file is able to retrieve it), it can be assumed that the target is in some way participating in the Peer-to-Peer network and steps can be taken to remove the Peer-to-Peer network software application.

In another embodiment, a software program is executed (**500**) on a computer system that has access to one or more target systems' file systems. The sofiware program creates (**501**) a target file. The data contained within this target file is information that can be used to identify the target computer(s). This is useful when there are more than one computer systems being targeted, and optionally one filename is used for ease of searching. The target file is placed into the folders of the target computer(s). For example, the target file is placed in as many folders as possible because the "shared folder" on each target computer is not known. Where there is more than one computer system being targeted, the data contained within this target file is optionally varied for each target computer. A search is initiated (**503**) on a Peer-to-Peer network (e.g., from a scanner system that connects (**502**) to the Peer-to-Peer network) to check (**504**) for the presence of the target file. If the target file is located (**506**), the file is then acquired (**507**) and the data reviewed (**508**) to identify or notify (**509**) the corresponding target computer.

In another embodiment, a software program is executed on the target computer. The software program creates a target file. The data contained within this target file is information that can be used to identify the target computer. This is useful when there are more than one computer system being targeted, and optionally one filename is used for ease of searching. The data that this target file includes is encrypted to protect the contents. The target file is placed into the folders of the target computer(s). For example, the target file is placed in as many folders as possible because the "shared folder" on each target computer is not known. Where there is more than one computer system being targeted, the data contained within this target file is optionally varied for each target computer. A search is initiated on a Peer-to-Peer network to check for the presence of the target file. If the target file is located (e.g., a node searching for the file is able to retrieve it), the file is then acquired. Once the file is acquired, the data contained within it is decrypted and reviewed to identify the corresponding target computer.

In another embodiment, a software program is executed on a computer system that has access to a target system's file systems. The software program creates a target file. The target file is placed into the folders of the target computer (i.e., the computer that is to be monitored for the purpose of detecting whether the target computer contains Peer-to-Peer network application software). For example, the target file is placed in as many folders as possible because the "shared folder" is not known. A search is initiated on a Peer-to-Peer network to check for the presence of the target file. If the target file is located (e.g., a node searching for the file is able to retrieve it), it can be assumed that the target computer is in some way participating in the Peer-to-Peer network and steps can be taken to remove the Peer-to-Peer network software application from the target computer.

In another embodiment, a target file is placed into the folders of the target computer. The target file is placed in as many folders as possible because the "shared folder" is not known. A search is initiated on a Peer-to-Peer network to check for the presence of the target file. If the target file is located (e.g., a node searching for the file is able to retrieve it), it can be assumed that the target is in some way participating in the Peer-to-Peer network and steps can be taken to remove the Peer-to-Peer network software application from the target computer.

In another embodiment, a software program is executed on a computer system that has access to one or more target systems' file systems. The software program creates a target file. The data contained within this target file is information that can be used to identify the target computer(s). This is useful when there are more than one computer systems being targeted, and optionally one filename is used for ease of searching. The target file is placed into the folders of the target computer(s). For example, the target file is placed in as many folders as possible because the "shared folder" on each target computer is not known. Where there is more than one computer system being targeted, the data contained within this target file is optionally varied for each target computer. A search is initiated on a Peer-to-Peer network to check for the presence of the target file. If the target file is located, the file is then acquired and the data reviewed to identify the corresponding target computer.

In another embodiment, a software program is executed on a computer system that has access to one or more target systems' file systems. The software program creates a target file. The data contained within this target file is information that can be used to identify the target computer(s). This is useful when there are more than one computer system being targeted, and optionally one filename is used for ease of searching. The data that this target file includes is encrypted to protect the contents. The target file is placed into the folders of the target computer(s). For example, the target file is placed in as many folders as possible because the "shared folder" on each target computer is not known. Where there

is more than one computer system being targeted, the data contained within this target file is optionally varied for each target computer. A search is initiated on a Peer-to-Peer network to check for the presence of the target file. If the target file is located, the file is then acquired. Once the file is acquired, the data contained within it is decrypted and reviewed to identify the corresponding target computer.

In yet another embodiment, a target file is placed onto a target system and a search is initiated via the peer-to-peer network for the target file. If the file is detected notification occurs. For example, an administrator responsible for monitoring the target computer is sent an electronic communication informing the administrator that the target computer is operating a peer-to-peer network software application.

In yet another embodiment, a target file is placed onto a target system. A monitoring agent is placed between the target system and the peer-to-peer network. A search is initiated via the peer-to-peer network for the target file. If the file is detected by the monitoring agent, then notification occurs. For example, an administrator responsible for monitoring the target computer is sent an electronic communication informing the administrator that the target computer is operating a peer-to-peer network software application. Optionally the monitoring agent automatically disables access to the node that has peer-to-peer software (i.e., the target system).

In yet another embodiment, a monitoring agent is placed between the target system and the peer-to-peer network. A search is initiated for a specific term via the peer-to-peer network. The monitoring agent is configured to monitor data inbound to nodes it is configured to protect (e.g., the target system). If the monitoring agent detects the specific term it will assume that the target system has peer-to-peer software, and automatically disable access to the target system.

In yet another embodiment, a monitoring agent is placed between the target computer and the peer-to-peer network. This monitoring agent could be a device on its own, embedded in a router or firewall, or other network device which passes network data. A software program is executed on the target computer. The software program creates a target file. The target file is placed into the folders of the target computer. For example, the target file is placed in as many folders as possible because the "shared folder" is not known. A search is initiated on a Peer-to-Peer network to check for the presence of the target file. If the monitoring agent detects the search string for the file the monitoring agent automatically blocks traffic to and from the target computer to prevent network access.

In yet another embodiment, a monitoring agent is placed between the target computer and the peer-to-peer network. This monitoring agent could be a device on its own, embedded in a router or firewall, or other network device which passes network data. A software program is executed on one or more target computers. The software program creates a target file. The data contained within this target file is information that can be used to identify the target computer. This is useful when there are more than one computer system being targeted, and optionally one filename is used for ease of searching. The data that this target file includes is encrypted to protect the contents. The target file is placed into the folders of the target computer(s). For example, the target file is placed in as many folders as possible because the "shared folder" is not known. Where there is more than one computer system being targeted, the data contained within this target file is optionally varied for each target computer. A search is initiated on a Peer-to-Peer network to check for the presence of the target file. If the target file is located, the file is then acquired. Once the file is acquired, the data contained within it is decrypted and reviewed to identify the corresponding target computer. The searching system then notifies the monitoring agent which automatically blocks traffic to and from the identified target computer (s) to prevent network access.

In yet another embodiment, a corporate network is protected by placing a target file in the folders of the computers located on the network, optionally changing the name of each file, or optionally encrypting the data contained within it. The file(s) are searched for on a Peer-to-Peer network, and if the file is detected it is optionally downloaded. Administrators or the user could then be notified.

In one embodiment, the invention is implemented in a computer system that contains a processor unit, main memory, and an interconnect bus. The processor unit may contain a single microprocessor, or may contain a plurality of microprocessors for configuring the computer as a multiprocessor system. The main memory stores, in part, instructions and data for execution by the processor unit. If the ability of the inventive system is wholly or partially implemented in software, the main memory may be used to store the executable code when in operation. The main memory may include banks of dynamic random access memory as well as high speed memory.

The computer system may further include a mass storage device, peripheral devices, portable storage medium drives, input control device, a graphics subsystem, and an output display. The computer system may be connected through one or more data transport means. For example, the processor unit and the main memory may be connected via a local microprocessor bus, and the mass storage device, peripheral devices, portable storage medium drives, graphics subsystem may be connected via one or more input/output (I/O) busses. The mass storage device, which may be implemented with a magnetic disk drive or an optical disk drive, is nonvolatile storage device for storing data and instructions for use by the processor unit. In a software embodiment, the mass storage device stores the software for loading to the main memory.

The input control device(s) provide a portion of the user interface for a user of the computer system. The input control devices may include an alpha numeric keypad for inputting alphanumeric and other key information, a cursor control device, such as a mouse, a trackball, a stylus, or cursor direction keys. In order to display textual and graphical information, the computer system contains the graphics subsystem and the output display. The output display may include a cathode ray tube display or a liquid crystal display. The graphics subsystem receives textual and graphical information and processes the information for output to the output display.

The components contained in the computer system are those typically found in general purpose computer systems, and in fact, these components are intended to represent a broad category of such computer components that are well known in the art.

The system may be implemented in either hardware or software. For some software embodiments, the software includes a plurality of computer executable instructions for implementation on a general purpose computer system. Prior to loading into a general purpose computer system, the system may reside as encoded information on a computer readable medium, such as a magnetic floppy disc, magnetic tape compact disc read only memory (CD-ROM). In one hardware embodiment, the system may comprise a dedicated processor including processor instructions for per-

forming the functions described herein. Circuits may also be developed to perform the functions described herein.

## EXAMPLES

The following Examples illustrate various embodiments of systems according to the present Invention.

### Example 1

This example illustrates a system for detecting Peer-to-Peer software applications by creating a specific target file, placing this target file in the folders of a target computer, and searching for this target file on a Peer-to-Peer network.

In this example, a user has installed a Peer-to-Peer software application onto computer system #**1**. A network administrator wishes to identify if this computer system has a Peer-to-Peer software application installed. The network administrator executes the detection software. The detection software creates a file named, "123456.txt" and places this file in every folder of computer system #**1**. The network administrator then issues a search on a Peer-to-Peer network for "123456.txt." The network administrator locates a file named, "123456.txt." The network administrator now knows that computer system #**1** has a Peer-to-Peer network software application installed.

### Example 2

This example illustrates a system for a detecting Peer-to-Peer software application by creating a specific target file with specific data, placing this target file in the folders of a target computer, and searching for this target file on a Peer-to-Peer network.

In this example, a user has installed a Peer-to-Peer software application onto computer system #**1** which has an IP address of 192.168.0.1. A network administrator wishes to identify if this computer system has a Peer-to-Peer software application installed thereon. The network administrator executes the detection software. The detection software creates a file named, "123456.txt." with the contents of this file being the IP address of computer system #**1**. The detection software places this file in every folder of computer system #**1**. The network administrator then issues a search on a Peer-to-Peer network for "123456.txt." The network administrator locates a file named, "123456.txt." The network administrator acquires the file and reviews the data. The IP address within the file is "192.168.0.1." The network administrator now knows that computer system #**1** has a Peer-to-Peer network software application installed thereon.

### Example 3

This example illustrates a system for detecting a Peer-to-Peer software application on multiple computer systems by creating a specific target file with specific data, placing this target file in the folders of a target computer, and searching for this target file on a Peer-to-Peer network.

In this example there are five computer systems on a network:

Computer system #**1** with an IP address of 192.168.0.1
Computer system #**2** with an IP address of 192.168.0.2
Computer system #**3** with an IP address of 192.168.0.3
Computer system #**4** with an IP address of 192.168.0.4
Computer system #**5** with an IP address of 192.168.0.5

The network administrator wishes to identify if any of these computer systems have a Peer-to-Peer network software application installed thereon. One or more users have installed a Peer-to-Peer software application onto computer system #**1** and computer system #**3**. The network administrator executes the detection software on each computer system. The detection software on each computer system creates a file named, "12456.txt." with the contents of this file being the IP address of the corresponding computer system. The detection software places this file in every folder of the corresponding computer system. The network administrator then issues a search on a Peer-to-Peer network for "12456.txt." The network administrator locates two files named, "12456.txt." The network administrator acquires these file and reviews the data. The IP address within file #**1** is "192.168.0.1" and the IP address within file #**2** is "192.168.0.3." The network administrator now knows that computer system #**1** and computer system #**3** have a Peer-to-Peer network software application installed thereon.

### Example 4

This example illustrates a system for detecting a Peer-to-Peer software application on the computers of a network which share the same IP addresses by creating a specific target file with specific data, placing this file in the folders of a target computer, and searching for this file on a Peer-to-Peer network.

In this example there is a corporate network that includes two remote offices. Each remote office network has two computer systems. Each computer system has a unique computer name. Each remote office utilizes an IP address scheme that is the same as the other. The resulting IP addresses are:

Remote Office #**1**, Computer System #**1**: COMPA, 192.168.0.1
Remote Office #**1**, Computer System #**2**: COMPB, 192.168.0.2
Remote Office #**2**, Computer System #**1**: COMPC, 192.168.0.1
Remote Office #**2**, Computer System #**2**: COMPD, 192.168.0.2

One or more users have installed a Peer-to-Peer software application on computer system #**1** in remote office #**1** and on computer system #**2** in remote office #**2**. A network administrator wishes to identify if any computer system on either remote office network has a Peer-to-Peer software application installed thereon. The network administrator executes the detection software on all computer systems on both remote office networks. The detection software on each computer system creates a file named, "12456.txt." with the contents of this file being the IP address and name of the corresponding computer system. The detection software places this file in every folder of the corresponding computer system. The network administrator then issues a search on a Peer-to-Peer network for "12456.txt." The network administrator locates two files named, "12456.txt." The network administrator acquires these file and reviews the data. The IP address within file #**1** is "192.168.0.1" and the IP address within file #**2** is "192.168.0.2." The name in file #**1** is "COMPA" and the name in file #**2** is "COMPD." The network administrator now knows that computer system #**1** in remote office #**1** and computer system #**2** in remote office #**2** have a Peer-to-Peer network software application installed thereon.

### Example 5

This example illustrates a secure system for detecting a Peer-to-Peer software application on multiple computer sys-

tems by creating a specific target file with specific data, encrypting this data, placing this file in the folders of a target computer, and searching for this file on a Peer-to-Peer network.

In this example, there are five computer systems on a network:

Computer system #**1** with an IP address of 192.168.0.1
Computer system #**2** with an IP address of 192.168.0.2
Computer system #**3** with an IP address of 192.168.0.3
Computer system #**4** with an IP address of 192.168.0.4
Computer system #**5** with an IP address of 192.168.0.5

The network administrator wishes to identify in a secure manner if any of these computer systems have a Peer-to-Peer network software application installed thereon. One or more users have installed a Peer-to-Peer software application onto computer system #**1** and computer system #**3**. The network administrator executes the detection software on each computer system. The detection software on each computer system creates a file named, "12456.txt." with the contents of this file being the IP address of the corresponding computer system. The detection software encrypts the contents of the file. The detection software places this file in every folder of the corresponding computer system. The network administrator then issues a search on a Peer-to-Peer network for "12456.txt." The network administrator locates two files named, "12456.txt." The network administrator acquires these files, decrypts the data, and reviews the data. The IP address within file #**1** is "192.168.0.1" and the IP address within file #**2** is "192.168.0.3." The network administrator now knows that computer system #**1** and computer system #**3** have a Peer-to-Peer network software application installed thereon.

Finally, it will be appreciated by those skilled in the art that changes could be made to the embodiments described above without departing from the broad inventive concept thereof. It is understood, therefore, that this invention is not limited to the particular embodiments disclosed, but is intended to cover modifications within the spirit and scope of the present invention as defined in the appended claims.

What is claimed is:

**1**. A method for detecting *whether* peer-to-peer **[**network**]** *file sharing* software *is* operating on a target computer, *the target computer connected to a computer network,* comprising:

**[**a.**]** creating, *by a software program configured to access a file system of the target computer,* a target file**[**, and placing the target file**]** *that contains identifying data* in one or more folders on the target computer, *the identifying data uniquely identifying the target computer*;

**[**b. locating**]** *receiving, by the software program, a search string of a search for* the target file on a **[**Peer-to-Peer**]** *peer-to-peer file sharing* network, *wherein the search is issued from a node of the peer-to-peer file sharing network to locate the target file* independent of the target computer responding to **[**a**]** *the* search issued on the **[**Peer-to-Peer**]** *peer-to-peer* network for the target file; **[**and

c. detecting a**]** *determining, by the software program, that the target file is shared to one or more nodes of the* peer-to-peer *file sharing network as a result of the search;*

*determining, by the software program, that the peer-to-peer file sharing* software *is* operating on the target computer **[**in an event the target file is located on the Peer-to-Peer network**]** *in response to determining that the target file is in at least one folder of the target computer; and*

*blocking, by the software program, the target computer from communicating data traffic with the peer-to-peer file sharing network.*

**2**. The method of claim **1**, wherein the target file is placed in a plurality of folders on the target computer.

**[3**. The method of claim **1**, wherein the target file contains data that uniquely identifies the target computer.**]**

**4**. The method of claim **[3]** *1*, wherein the *identifying* data is encrypted.

**5**. The method of claim **[3]** *1*, wherein the *identifying* data includes *at least one of* an IP address of the target computer, *a name of the target computer, a name of a user of the target computer, an email address of the user of the target computer, or information entered by the network administrator or operator responsible for monitoring the target computer*.

**[6**. The method of claim **3**, wherein the data contains a name of the target computer.**]**

**[7**. The method of claim **3**, wherein the data contains a name of a user of the target computer.**]**

**[8**. The method of claim **3**, wherein the data contains an email address of a user of the target computer.**]**

**[9**. The method of claim **3**, wherein the data contains information entered by a network administrator or operator responsible for monitoring the target computer.**]**

**[10**. The method of claim **1**, wherein the method is implemented at least in part using software that executes on the target computer.**]**

**11**. The method of claim **1**, wherein the method is implemented at least in part using software that executes on a computer different from the target computer.

**12**. The method of claim **1**, further comprising: notifying at least one of a firewall, an intrusion detection system, a router, or an application, that *the* peer-to-peer *file sharing* software has been **[**detected**]** *determined to be operating on the target computer*, wherein the notification is performed based in accordance with an output of the **[**detecting**]** *determining* step.

**13**. A system for detecting *whether* peer-to-peer **[**network**]** *file sharing* software *is* operating on a target computer, *the target computer connected to a computer network,* comprising:

a storage medium for storing instructions; **[**and**]**

a user input device for receiving user input; and

a processor unit operable to process the user input and to use the instructions to:

create a target file**[**; place the target file**]** in one or more folders *of the storage medium* on the target computer *prior to an administrator knowing whether or not the target computer is participating in peer-to-peer file sharing*;

**[**locate**]** *receive a search string of a search for* the target file on a **[**Peer-to-Peer**]** *peer-to-peer file sharing* network, *wherein the search is issued from a node of the peer-to-peer file sharing network to locate the target file* independent of the target computer responding to **[**a**]** *the* search issued on a **[**Peer-to-Peer**]** *peer-to-peer* network for the target file; **[**and**]**

*determine that the target file is shared to one or more nodes of the peer-to-peer file sharing network as a result of the search;*

**[**detect whether**]** *determine that the* peer-to-peer *file sharing* software is operating on the target computer **[**in an event the target file is located on the Peer-to-Peer network**]** *in response to determining that the target file is in at least one folder of the target computer; and*

*block the target computer from communicating data traffic with the peer-to-peer file sharing network.*

**14**. The system of claim **13**, wherein the processor unit is operable to process the user input and to use the instructions to place the target file in a plurality of folders on the target computer.

**[15**. The system of claim **14**, wherein the target file contains data that uniquely identifies the target computer.**]**

**16**. The system of claim **[15]** *13*, wherein the *identifying* data is encrypted.

**17**. The system of claim **[15]** *13*, wherein the *identifying* data includes *at least one of* an IP address of the target computer, *a name of the target computer, a name of a user of the target computer, an email address of the user of the target computer, information entered by a network administrator, or an operator responsible for monitoring the target computer*.

**[18**. The system of claim **15**, wherein the data contains a name of the target computer.**]**

**[19**. The system of claim **15**, wherein the data contains a name of a user of the target computer.**]**

**[20**. The system of claim **15**, wherein the data contains an email address of a user of the target computer.**]**

**[21**. The system of claim **15**, wherein the data contains information entered by a network operator or administrator responsible for monitoring the target computer.**]**

**[22**. The system of claim **13**, wherein the system is implemented using software that executes at least in part on the target computer.**]**

**[23**. The system of claim **13**, wherein the system is implemented using software that executes at least in part on a computer different from the target computer.**]**

**24**. The system of claim **13**, wherein the processor unit automatically notifies at least one of a firewall, an intrusion detection system, a router, or an application, upon detection of peer-to-peer software.

**[25**. A system for the detection of one or more Peer-to-Peer network software applications operating on a target computer, comprising:

a. means for creating a target file, and placing the target file in one or more folders on the target computer;

b. means for locating the target file on a Peer-to-Peer network independent of the target computer responding to a search issued on a Peer-to-Peer network for the target file; and

c. means for detecting whether peer-to-peer software is operating on the target computer in an event the target file is located on the Peer-to-Peer network.**]**

**[26**. The system of claim **25**, wherein the target file is placed in a plurality of folders on the target computer.**]**

**[27**. The system of claim **25**, wherein the target file contains data that uniquely identifies the target computer.**]**

**[28**. A method executed by a monitoring agent on a peer-to-peer network for detecting peer-to-peer network participation of a first node, comprising:

a. monitoring network data sent to the first node; and

b. issuing a search for a predetermined term on a peer-to-peer network while monitoring the network data sent to the first node;

c. detecting peer-to-peer participation of the first node in the peer-to-peer network if the monitoring identifies the the search being sent to the first node, the detecting being independent of the first node receiving the issued search.**]**

**[29**. The method of claim **28**, further comprising notifying an administrator responsible for monitoring the node if peer-to-peer participation is detected.**]**

**[30**. The method of claim **28**, further comprising automatically blocking data access to the first node if peer-to-peer participation is detected.**]**

**[31**. The method of claim **28**, further comprising disabling peer-to-peer software associated with the first node if peer-to-peer participation is detected.**]**

**[32**. The method of claim **28**, wherein the first node resides on a corporate computer network.**]**

**[33**. The method of claim **28**, wherein the first node resides on a home computer network.**]**

**[34**. A system for detecting peer-to-peer network participation, comprising:

a. means for monitoring network data sent to the first node;

b. means for issuing a search for a predetermined term on a peer-to-peer network while the means for monitoring the network data monitors the network data sent to the first node; and

c. means for detecting peer-to-peer participation of the first node in the peer-to-peer network if the means for monitoring identifies the the search being sent to the first node, the detecting means being independent of the first node receiving the issued search.**]**

*35. A method comprising:*

*receiving, by a target computer, a target file to be placed in one or more folders on the target computer, the target file comprising identifying data, the identifying data uniquely identifying the target computer, wherein the target computer is configured to transmit and receive one or more messages over the computer network;*

*receiving, by target computer, a search string of a search on a peer-to-peer file sharing network for the target file;*

*determining, by the target computer, that the target file is on the peer-to-peer file sharing network in response to issuing the search, wherein the search is issued from a node of the peer-to-peer file sharing network to locate the target file independent of the target computer responding to the search issued on the peer-to-peer network for the target file;*

*determining, by the target computer, that peer-to-peer file sharing software is operating on the target computer in response to determining that the target file is in at least one folder of the target computer; and*

*blocking, by the target computer, the target computer from communicating data traffic with the one or more nodes of the peer-to-peer file sharing network in response to determining that the peer-to-peer file sharing software is on the target computer.*

*36. The method of claim 35, wherein the target file is placed in one or more folders located on the target computer.*

*37. The method of claim 35, wherein the identifying data includes at least one of an IP address of the target computer, a name of the target computer, or a name of a user of the target computer.*

*38. The method of claim 37, further comprising, removing the peer-to-peer software from the target computer.*

*39. A method for detecting peer-to-peer file sharing software operating on a target computer participating in a computer network, comprising:*

*creating, by a software program of a computer, a target file containing data uniquely identifying the target computer in one or more folders on the target computer prior to an administrator knowing whether or not the target computer is participating in a peer-to-peer file sharing network;*

*issuing, by the software program, a search for the target file on the peer-to-peer file sharing network, wherein the search is issued to locate the target file independent of the target computer responding to the search issued on the peer-to-peer network for the target file;*

*determining, by the software program, that the peer-to-peer file sharing software is operating on the target computer based upon the search results received from the target computer; and*

*blocking, by the software program, the target computer from communicating data traffic in the peer-to-peer file sharing network.*

*40. The method of claim 39, wherein the target file is placed in a plurality of folders on the target computer.*

*41. A system for detecting peer-to-peer file sharing software operating on a target computer participating in a computer network, comprising:*

*a storage medium for storing instructions; and*

*a user input device for receiving user input; and*

*a processor unit operable to process the user input and to use the instructions to:*

*create a target file containing data uniquely identifying the target computer in one or more folders on the target computer prior to an administrator knowing whether or not the target computer is participating in a peer-to-peer file sharing network;*

*issue a search for the target file on the peer-to-peer file sharing network, wherein the search is issued to locate the target file independent of the target computer responding to the search issued on the peer-to-peer network for the target file;*

*determine that the peer-to-peer file sharing software is operating on the target computer based upon the search results received from the target computer; and*

*block the target computer from communicating data traffic in the peer-to-peer file sharing network.*

*42. The system of claim 41, wherein the processor unit is operable to process the user input and to use the instructions to place the target file in a plurality of folders on the target computer.*

* * * * *