



(51) International Patent Classification:
G06F 21/51 (2013.01) *H04L 29/06* (2006.01)
G06F 21/60 (2013.01)

(21) International Application Number:
PCT/MY2015/050071

(22) International Filing Date:
13 July 2015 (13.07.2015)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
PI2014002188 24 July 2014 (24.07.2014) MY

(71) Applicant: MIMOS BERHAD [MY/MY]; Technology Park Malaysia, Kuala Lumpur, 57000 (MY).

(72) Inventors: MAT NOR, Fazli; c/o Mimos Berhad, Technology Park Malaysia, Kuala Lumpur, 57000 (MY). BALAKRISHNAN, Nathan M; c/o Mimos Berhad, Technology Park Malaysia, Kuala Lumpur, 57000 (MY). ABD AZIZ, Mohd Azrul Hadi; c/o Mimos Berhad, Technology Park Malaysia, Kuala Lumpur, 57000 (MY). AHMAD, Nurul Haszeli; c/o Mimos Berhad, Technology Park Malaysia, Kuala Lumpur, 57000 (MY). M. PERUMAL, Nagendran; c/o Mimos Berhad, Technology Park Malaysia, Kuala Lumpur, 57000 (MY).

(74) Agent: MIRANDAH ASIA (MALAYSIA) SDN BHD; Suite 3B-19-3, Plaza Sentral, Jalan Stesen Sentral 5, Kuala Lumpur, 50470 (MY).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

Declarations under Rule 4.17:

— as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))

[Continued on next page]

(54) Title: A SYSTEM AND METHOD TO SECURE DISTRIBUTION AND EXECUTION OF LIBRARY FILES IN A CLIENT-SERVER ENVIRONMENT

100

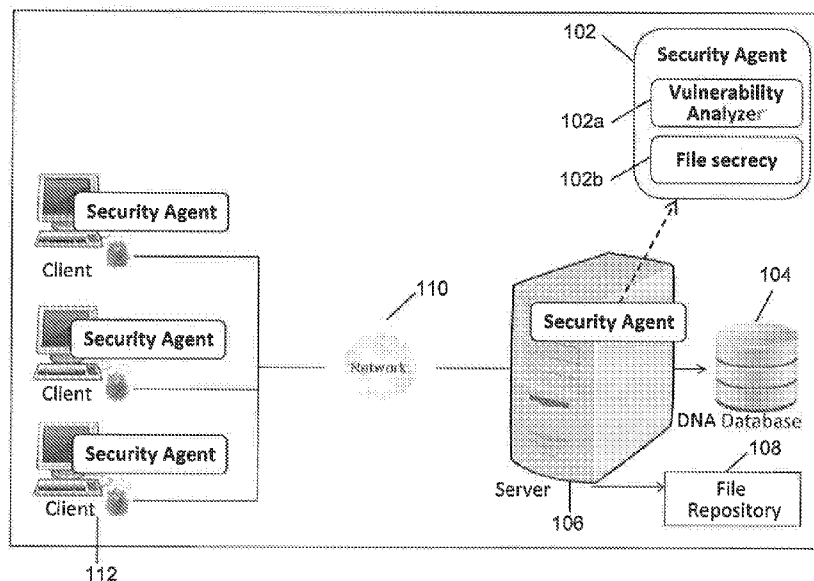


FIG. 1.0

(57) Abstract: The system and method of the present invention secures distribution and execution of library files in a client-server environment by splitting through random number and key generator which prevents file from being tampered and program analysis which prevents exploitation of file. The present invention comprising at least one server (106); at least one client device (112) at client side; and at least a database (104). The server (106) for hosting library file further comprises at least one Security Agent (102) for performing file vulnerability analysis, file secrecy and updating the database with file DNA (Deoxyribonucleic acid). Generally, the present invention encompassed fragmentation of executable file to multiple chunks using file size and random number; file security check using file chunk mapping information and file DNA; and file vulnerability using malware analysis before distribution and execution.



— *of inventorship (Rule 4.17(iv))*

Published:

— *with international search report (Art. 21(3))*

A SYSTEM AND METHOD TO SECURE DISTRIBUTION AND EXECUTION OF LIBRARY FILES IN A CLIENT-SERVER ENVIRONMENT

FIELD OF INVENTION

5

The present invention relates to a system and method for secure distribution and execution of library files in a client-server environment. In particular, the invention relates to systems and methods that utilize file fragmentation based on random file size and chunk position which prevents files from being tampered and program analysis which prevents exploitation of files or harming the system.

10

BACKGROUND ART

15

Distribution and execution of files (i.e. documents, binaries, libraries, etc) are transferred between client(s) and server. These files are placed in well-defined set of directories in the server before it can be downloaded by the client.

20

Insecure process of file transfer may lead to security issues if files transferred between client and server does not undergo proper checking. The files may be exploited by attackers. Current security implementations such as Cryptography which is referred exclusively to encryption as well as trust and privacy application are not sufficient. There is a need to measure the contents automatically in use, in motion and in rest to ensure that the process and files are completely secured which would strengthen security of the environment.

25

30

United States Patent Publication No. US 2009/0252325 A1 (US '325 Publication) entitled Secure Content Pre-Distribution to Designated Systems discloses pre-distributing content to client recipients, such that the distribution is largely complete, but in which the content is encrypted and thus not able to be used until a key (or set of keys) is made available for decrypting the content. The US' 325 Publication provide a secure distribution and execution method. Further, content is published to distribution server before distributing to client. Package file is secured before publishing to client and package hash is calculated on the encrypted package to detect any untrusted

modification. US '325 Publication does not involve file splitting method and does not provide for any vulnerability analysis.

United States Patent No. US 8069349 B1 (US '349 Patent) entitled Method of secure file transfer relates to multi-computer data transferring and, in particular to computer-to-computer data transfer regulating. US '349 Patent provides a method of transferring files from untrusted network to trusted network. A series of initial checks (virus, file size and file extension) is performed and file elements are separated into constituent part and reassembling if no violation of security. Image security checks and text security checks is performed and the files will be deleted if the image security checks and text security checks are violated. Secure transfer of an archive file of constituent is provided as compared to the present invention which secures and fragments executable files to multiple chunks using file size and random number. In the invention as disclosed in the US '349 Patent, extraction to files is performed first and thereafter proceeds with file security check. In the present invention, file security check is performed first before fragmentation of files. Further, the invention as disclosed in US '349 Patent does not use random number for file extraction as compared to the present invention which utilizes random number for key generation for fragmentation.

The publication entitled "A Novel Encryption Scheme for Data Deduplication System" by Can Wang, Zhi-guang Qin, Jin Peng, Juan Wang; 2010 IEEE proposes an encryption scheme which transformed encryption from file to chunks. Further, security analysis is performed. However, said publication does not use random number for data deduplication as compared to the present invention which utilizes random number for key generation for fragmentation. Vulnerability analysis is not implemented in said publication as compared to the present invention which utilizes vulnerability analysis to detect malicious code embedded in file content.

The subject matter claimed herein is not limited to embodiments that solve any disadvantages or that operate only in environments such as those described above. Rather, this background is only provided to illustrate one exemplary technology area where some embodiments described herein may be practice.

SUMMARY OF INVENTION

The present invention relates to a system and method for secure distribution and execution of library files in a client-server environment. The invention relates to
5 utilization of file fragmentation based on random file size and chunk position which prevents from tampering of files and program analysis which prevents from exploitation of files which harms the system.

One aspect of the present invention provides a system to secure distribution and
10 execution of library files in a client-server environment. The system comprising at least one server for hosting library file; at least one client device at client side connecting to the server to download the library file; and at least a database for storing updates. The at least one server (106) for hosting library file further comprises at least one Security Agent for performing file vulnerability analysis, file secrecy and updating the database
15 with file DNA.

Another aspect of the present invention provides that the at least one Security Agent further comprises at least one Vulnerability Analyzer for analyzing vulnerabilities and integrity measurement; and at least one File Secrecy Module for securing file during
20 distribution.

A further aspect of the present invention provides a method to secure distribution and execution of library file in a client-server environment. The method comprising steps of
25 executing configuration process; invoking security mechanism based on the configuration process; executing vulnerability analysis, integrity measurement and secrecy process; distributing file to client device by server if no thread is found; downloading file from server by client; reversing file secrecy process by the Security Agent for execution at client side; executing program analysis and integrity measurement by the Security Agent; and executing said library file. The step of
30 executing vulnerability analysis, integrity measurement and secrecy process further comprises steps of uploading file to server; invoking Security Agent to load file into memory; performing vulnerability analysis on loaded file to ensure file is free from exploitable vulnerabilities and any malicious attack and storing DNA of program if file satisfy rules in program analysis before process is completed; determining if any file is

not identified as threat; performing file secrecy process by Security Agent by performing combination of encryption and file splitting process to produce new encrypted compress file when no file is identified as threat; storing said encrypted compress file into define repository by Security Agent; and sending file to client by the server when requested. If
5 any file is identified as threat, end process.

Yet another aspect of the present invention provides a method wherein performing vulnerability analysis on loaded file to ensure file is free from exploitable vulnerabilities and any malicious attack and storing DNA of program if file satisfy rules in program
10 analysis before process is completed further comprises steps of analyzing file by Security Agent for vulnerabilities; determining if vulnerabilities exceeds threshold based on configuration process; if vulnerabilities does not exceed threshold, generating DNA of file which is unique to the file by the Security Agent if vulnerabilities of said vulnerability analysis does not exceed the threshold level; analyzing file for malware behavior or
15 characteristics by Security Agent; determining if file is harmless; hashing DNA, storing result in defined DNA database by Security Agent; and identifying file as no threat; if file is harmful; notification will be sent by Security Agent and the file will be identified as threat; if vulnerabilities exceeds threshold, sending notification by Security Agent and identifying file as threat.

20

Still another aspect of the present invention provides a method wherein performing file secrecy process by Security Agent by performing combination of encryption and file splitting process to produce new encrypted compress file when no file is identified as threat further comprises steps of generating random file size for splitting based on
25 original file (Z) size; splitting original file based on generated file size and assigning random names (B1, B2,...Bn) to the split file; randomly change chunk position from (B1, B2,...Bn) to (C1, C2,...Cn); creating a metadata mapping file (X) by the Security Agent and storing in the server; generating X's mapping reference (Y) with fixed size (M) by Security Agent; and combining new position of chunk (C1, C2,...Cn) with X's mapping
30 reference (Y) and encrypting and renaming it to Z .

Another aspect of the present invention provides a method wherein executing said library file further comprises steps of downloading file by client from server repository and executing the file (Z); loading file (Z) into memory by Security Agent; decrypting the

file (Z) by the Security Agent; reading first M bytes to retrieve X's mapping reference (Y) by Security Agent; requesting mapping file (X) from server using Y as a reference; decrypting mapping file (X) to retrieve mapping details and rearranging chunk to original position; and performing vulnerability analysis on file .

5

A further aspect of the present invention provides a method wherein performing vulnerability analysis on file further comprises steps of analyzing file for vulnerabilities by Security Agent; determining if vulnerabilities exceed threshold; if vulnerabilities does not exceed threshold, generating DNA of file; analyzing file for malware behavior or characteristics by Security Agent; determining if file is harmless; if file is harmless, hashing DNA by Security Agent and comparing with hashed DNA stored in DNA database; determining if both DNA matches; executing file if DNA matched; informing user and obtaining permission to execute the file if both DNA does not match; determining if permission is granted; executing file if permission is granted else end process; if file is not harmless, sending notification to file executor and end process; if vulnerabilities exceed threshold, sending notification to file executor and end process.

10

15

20

The present invention consists of features and a combination of parts hereinafter fully described and illustrated in the accompanying drawings, it being understood that various changes in the details may be made without departing from the scope of the invention or sacrificing any of the advantages of the present invention.

BRIEF DESCRIPTION OF ACCOMPANYING DRAWINGS

To further clarify various aspects of some embodiments of the present invention, a more particular description of the invention will be rendered by references to specific
5 embodiments thereof, which are illustrated in the appended drawings. It is appreciated that these drawings depict only typical embodiments of the invention and are therefore not to be considered limiting of its scope. The invention will be described and explained with additional specificity and detail through the accompanying drawings in which:

10 FIG. 1.0 illustrates the general architecture of the present invention.

FIG. 2.0 is a flowchart illustrating the general methodology of the present invention.

15 FIG. 3.0 is a flowchart illustrating the steps for executing vulnerability analysis, integrity measurement and secrecy process.

20 FIG. 4.0 is a flowchart illustrating the steps for performing vulnerability analysis on loaded file to ensure file is free from exploitable vulnerabilities and any malicious attack and storing DNA of program if file satisfy rules in program analysis before process is completed.

25 FIG. 5.0 is a flowchart illustrating the steps for performing file secrecy process by Security Agent by performing combination of encryption and file splitting process to produce new encrypted compress file when no file is identified as threat.

FIG. 6.0 is a diagram illustrating the file secrecy method.

FIG. 7.0 is a flowchart illustrating the steps for executing said library file.

30 FIG. 8.0 is a flowchart illustrating the steps for performing vulnerability analysis on file.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention relates to a system and method for secure distribution and execution of library file in a client-server environment. In particular, the invention relates to systems and methods which utilizes file size and random number for fragmentation of executable file to multiple chunk, file security check using file chunk mapping information and file DNA (Deoxyribonucleic acid) as well as file vulnerability through malware analysis prior to distribution and execution.

Hereinafter, this specification will describe the present invention according to the preferred embodiments. It is to be understood that limiting the description to the preferred embodiments of the invention is merely to facilitate discussion of the present invention and it is envisioned without departing from the scope of the appended claims.

Referring to FIG. 1.0, a general architecture of the present invention is illustrated. As illustrated in FIG. 1.0, the system of the present invention comprising at least one server (106) for hosting library file; at least one client device (112) at client side connecting to the server (106) to download the library file; and at least a database (104) for storing updates. The at least one server (106) for hosting library file further comprises at least one Security Agent (102) for performing file vulnerability analysis, file secrecy and updating the database with file DNA. The at least one Security Agent (102) further comprises at least one Vulnerability Analyzer (102a) for analyzing vulnerabilities and integrity measurement; and at least one File Secrecy Module (102b) for securing file during distribution.

FIG. 2.0 is a flowchart illustrating the general methodology of the present invention. The method includes executing a configuration process (202) followed by invoking the security mechanism based on the configuration process (204). Upon invoking the security mechanism, vulnerability analysis, integrity measurement and secrecy process (206) is executed and subsequently file is distributed to the client device by the server if no thread is found (208). File is downloaded from the server by client (210) and the file secrecy process is reversed by the Security Agent for execution at client side (212). Thereafter, the program analysis and integrity measurement is executed by the Security Agent (214); and the library file is executed (216).

FIG. 3.0 illustrates in more detail the steps of executing vulnerability analysis, integrity measurement and secrecy process. As illustrated in FIG. 3.0, administrator uploads file to server (302) and Security Agent load the file into memory (304). Thereafter, the Security Agent will proceed to perform vulnerability analysis on loaded file to ensure file is free from exploitable vulnerabilities and any malicious attack and storing DNA of program if file satisfy the condition in the program analysis before completion of the process (306). It is further determined if any of the file is identified as threat (308). File secrecy process is performed by the Security Agent by performing combination of encryption and file splitting process to produce new encrypted compress file when no file is identified as threat (310). Thereafter, Security Agents stores the encrypted compress file into define repository by (312); and the encrypted compress file is sent to client by the server when requested (314).

FIG. 4.0 illustrates the details of the further stages of vulnerability analysis of file of step 306 of FIG. 3.0. As illustrated in FIG. 4.0, Security Agent will first analyze the file for vulnerabilities (402). The technique for vulnerability analysis depends on the technique specified in the policy of the configuration process of the system and is not limited to integer range, lexical, or data flow analysis. It is determined if vulnerabilities exceeds the threshold based on the configuration process (404). If vulnerabilities do not exceed the threshold level, DNA of the file which is unique to the file is generated by the Security Agent (406). DNA of the file can be and not limited to Intermediate Representation (IR), Abstract Syntax Tree (AST), and file metadata. Thereafter, Security Agent analyzes file for malware behavior or characteristics (408). The technique for analysis of file for malware behavior or characteristics can be any available technique and not limited to Genetic Algorithm (GA) and Function Hooking. It is further determined if the file is harmless (410). If the file successfully go through the analysis process (i.e. no harmless file found), Security Agent will proceed to hash the DNA of the file and store the results in defined DNA database (416) and the file is identified as no threat (416). If file is determined as harmful, Security Agent sends notification to the administrator (412) and identify file as threat (414).

If vulnerabilities exceed the threshold level, Security Agent will send the notification (412) to file distributor using any media. The said media is not limited to email or messaging. Thereafter, distribution process is terminated and the flow ends. If the file is

found to be harmless and the probability of harm exceeds the threshold level, Security Agent will send notification to file distributor, identifies the file as threat (414), terminate the distribution process and end the flow.

- 5 FIGs. 5.0 and 6.0 illustrate the detail of file secrecy process of file during distribution process. As illustrated in FIG. 5.0, Security Agent randomized using a random function. The random function utilized any available random function and not limited to pseudo-random number or quantum random number generator. Security Agent generates a random file size to be split based on the original file (Z) size (502). Security Agent splits
10 the original file based on the generated file size and assign random names (B1, B2,...Bn) to the split file (504) and Security Agent randomly change the chunk position of file (B1, B2,...Bn) to (C1, C2,...Cn) (506). Thereafter, Security Agent creates a metadata file (X), encrypt and store it in the server. Metadata mapping file contains original position and the size of the chunks (508). Subsequently, X's mapping reference
15 (Y) with fixed size (M) is generated by the Security Agent (510) and the new position of chunk (C1, C2,...Cn) are combined with X's mapping reference (Y) and further being encrypted and renamed to Z (512). X will be stored in the server using Y as its reference key and Y is obtained through the original filename and timestamp (M kb).
- 20 FIG. 7.0 is a flowchart illustrating the steps for executing said library file. As illustrated in FIG. 7.0, client downloads file from server repository and executes the file (Z) (702). Security Agent will load the file into memory if the request made was to execute the file (704). Security agent decrypts the file once the file has been loaded (706) and read first M bytes from Z to retrieve Y (708). Thereafter, Security Agent requests X from server
25 using Y as a reference (710) and subsequently Security Agent decrypts X to retrieve mapping details and rearrange chunk to original position (712). Security Agent performs vulnerability analysis on the file (714).

- 30 The details of the further stages of vulnerability analysis of file of step 714 are described in detail in FIG. 8.0. The process of vulnerability analysis or program analysis during the execution process is similar to the process of program analysis during distribution in which vulnerability analysis will be performed. As illustrated in FIG. 8.0, vulnerability analysis is performed on file by analyzing file for vulnerabilities by Security Agent (802). It is determined if vulnerabilities exceed threshold (804). If vulnerabilities does not

exceed threshold, DNA of file is generated by Security Agent (808) and file is analyze for malware behavior or characteristics by the Security Agent (810). It is further determined if the file is harmless (812). If the file is harmless, Security Agent will proceed to hash the DNA of the file and compare it with the hashed DNA stored in the DNA database (814).

- 5 It is further determined if both the DNA matches (816) and executes the file if DNA matched (818). User is informed and permission is obtained to execute the file if both DNA does not match (820) and it is further determined if permission is granted (822). File is executed if permission is granted else the process ends (824). If file is not harmless, notification is sent to the file executor and process ends (806). If Security
- 10 Agent found that vulnerabilities exceed the threshold level or if malware characteristics are found, notification will be sent to the file executor and the process ends (806).

The present invention utilizes splitting method by using file fragmentation based on random file size and chunk position which prevents from file being tampered and

15 program analysis technique to prevent exploitation of file or harming of the computer system. The present invention provides for fragmentation of executable file to multiple chunks using file size and random number, file security check using file chunk mapping information and file DNA, and file vulnerability using malware analysis before distribution and execution.

20

Unless the context requires otherwise or specifically stated to the contrary, integers, steps or elements of the invention recited herein as singular integers, steps or elements clearly encompass both singular and plural forms of the recited integers, steps or elements.

25

Throughout this specification, unless the context requires otherwise, the word "comprise", or variations such as "comprises" or "comprising", will be understood to imply the inclusion of a stated step or element or integer or group of steps or elements or integers, but not the exclusion of any other step or element or integer or group of steps,

30 elements or integers. Thus, in the context of this specification, the term "comprising" is used in an inclusive sense and thus should be understood as meaning "including principally, but not necessarily solely".

It will be appreciated that the foregoing description has been given by way of illustrative example of the invention and that all such modifications and variations thereto as would be apparent to persons of skill in the art are deemed to fall within the broad scope and ambit of the invention as herein set forth.

CLAIMS

1. A system (100) to secure distribution and execution of library files in a client-server environment comprising:
 - 5 at least one server (106) for hosting library file;
 - at least one client device (112) at client side connecting to the server (106) to download the library file; and
 - at least a database (104) for storing updatescharacterized in that
 - 10 the at least one server (106) for hosting library file further comprises at least one Security Agent (102) for performing file vulnerability analysis, file secrecy and updating the database with file DNA (Deoxyribonucleic acid).
- 15 2. A system (100) according to Claim 1, wherein the at least one Security Agent (102) further comprises:
 - at least one Vulnerability Analyzer (102a) for analyzing vulnerabilities and integrity measurement; and
 - at least one File Secrecy Module (102b) for securing file during20 distribution.
3. A method (200) to secure distribution and execution of library file in a client-server environment comprising steps of:
 - executing configuration process (202);
 - 25 invoking security mechanism based on the configuration process (204);
 - executing vulnerability analysis, integrity measurement and secrecy process (206);
 - distributing file to client device by server if no thread is found (208);
 - downloading file from server by client (210);
 - 30 reversing file secrecy process by the Security Agent for execution at client side (212);
 - executing program analysis and integrity measurement by the Security Agent (214); and
 - executing said library file (216)

characterized in that
executing vulnerability analysis, integrity measurement and secrecy
process further comprises steps of:

- 5 uploading file to server (302);
- invoking Security Agent to load file into memory (304);
- performing vulnerability analysis on loaded file to ensure file is
 free from exploitable vulnerabilities and any malicious attack and
 storing DNA of program if file satisfy rules in program analysis
 before process is completed (306);
- 10 determining if any file is identified as threat (308);
- performing file secrecy process by Security Agent by performing
 combination of encryption and file splitting process to produce
 new encrypted compress file when no file is identified as threat
 (310);
- 15 storing said encrypted compress file into define repository by
 Security Agent (312); and
- sending file to client by the server when requested (314).

- 4. A method (400) according to Claim 3, wherein performing vulnerability analysis
20 on loaded file to ensure file is free from exploitable vulnerabilities and any
 malicious attack and storing DNA of program if file satisfy rules in program
 analysis before process is completed further comprises steps of:
 - analyzing file by Security Agent for vulnerabilities (402);
 - determining if vulnerabilities exceeds threshold based on configuration
25 process (404);
 - if vulnerabilities does not exceed threshold:
 - generating DNA of file which is unique to the file by the Security
 Agent (406);
 - analyzing file for malware behavior or characteristics by Security
30 Agent (408);
 - determining if file is harmless (410);
 - hashing DNA, storing result in defined DNA database
 by Security Agent (416); and identifying file as no threat
 (418);

sending notification by Security Agent (412) and identifying file as threat if file is harmful(414)

if vulnerabilities exceeds threshold:

sending notification by Security Agent and identifying file as threat (412,414).

5

5. A method (500) according to Claim 3, wherein performing file secrecy process by Security Agent by performing combination of encryption and file splitting process to produce new encrypted compress file when no file is identified as threat further comprises steps of:

10

generating random file size for splitting based on original file (Z) size (502);

splitting original file based on generated file size and assigning random names (B1, B2,...Bn) to the split file (504);

15

randomly change chunk position from (B1, B2,..Bn) to (C1, C2,...Cn) (506);

creating a metadata mapping file (X) by the Security Agent and storing in the server (508);

20

generating X's mapping reference (Y) with fixed size (M) by Security Agent (510); and

combining new position of chunk (C1, C2,...Cn) with X's mapping reference (Y) and encrypting and renaming it to Z (512).

6. A method (700) according to Claim 3, wherein executing said library file further comprises steps of:

25

downloading file by client from server repository and executing the file (Z) (702);

loading file (Z) into memory by Security Agent (704);

decrypting the file (Z) by the Security Agent (706);

30

reading M bytes from Z to retrieve Y's mapping reference by Security Agent (708);

requesting mapping file (X) from server using Y as a reference (710);

decrypting mapping file (X) to retrieve mapping details and rearranging chunk to original position (712); and

performing vulnerability analysis on file (714).

5 7. A method according to Claim 3, wherein executing configuration process; said configuration process includes policy setup, encryption, random number generation, program analyzer and file splitting setting.

8. A method (800) according to Claim 6, wherein performing vulnerability analysis on file further comprises steps of:

analyzing file for vulnerabilities by Security Agent (802);

10 determining if vulnerabilities exceed threshold (804);

if vulnerabilities does not exceed threshold:

generating DNA of file (808);

analyzing file for malware behavior or characteristics by Security Agent (810);

15 determining if file is harmless (812);

if file is harmless:

hashing DNA by Security Agent and comparing with hashed DNA stored in DNA database (814);

determining if both DNA matches (816);

20 executing file if DNA matched (818);

informing user and obtaining permission to execute the file if both DNA does not match (820);

determining if permission is granted (822);

25 executing file if permission is granted else end process (824)

if file is not harmless:

sending notification to file executor and end process (806)

if vulnerabilities exceed threshold:

sending notification to file executor and end process (806).

30

9. A method according to Claim 3, wherein said vulnerability analysis depends on technique specified in policy setup of the configuration process and is not limited to integer range, lexical or data flow analysis.

10. A method according to Claim 5, wherein metadata mapping file (X) contains original position and size of the chunks.

100

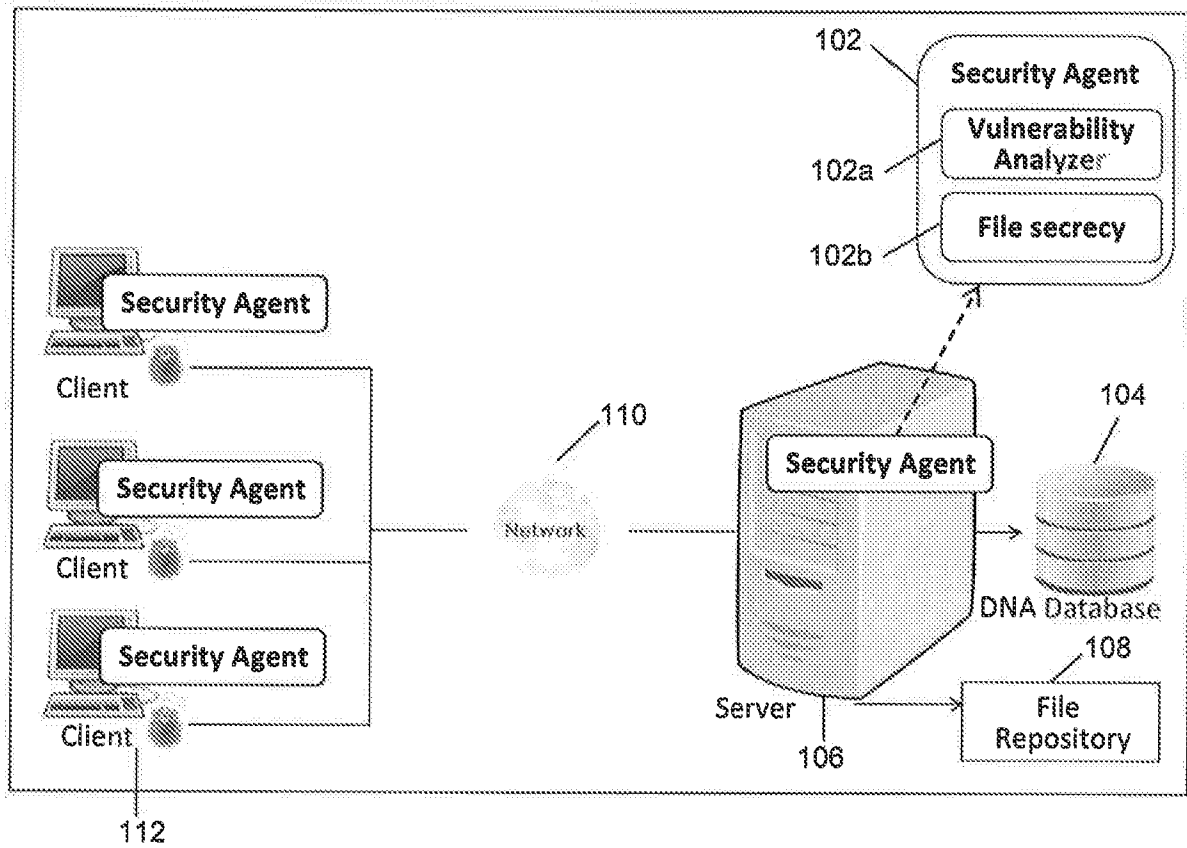


FIG. 1.0

2/8

200

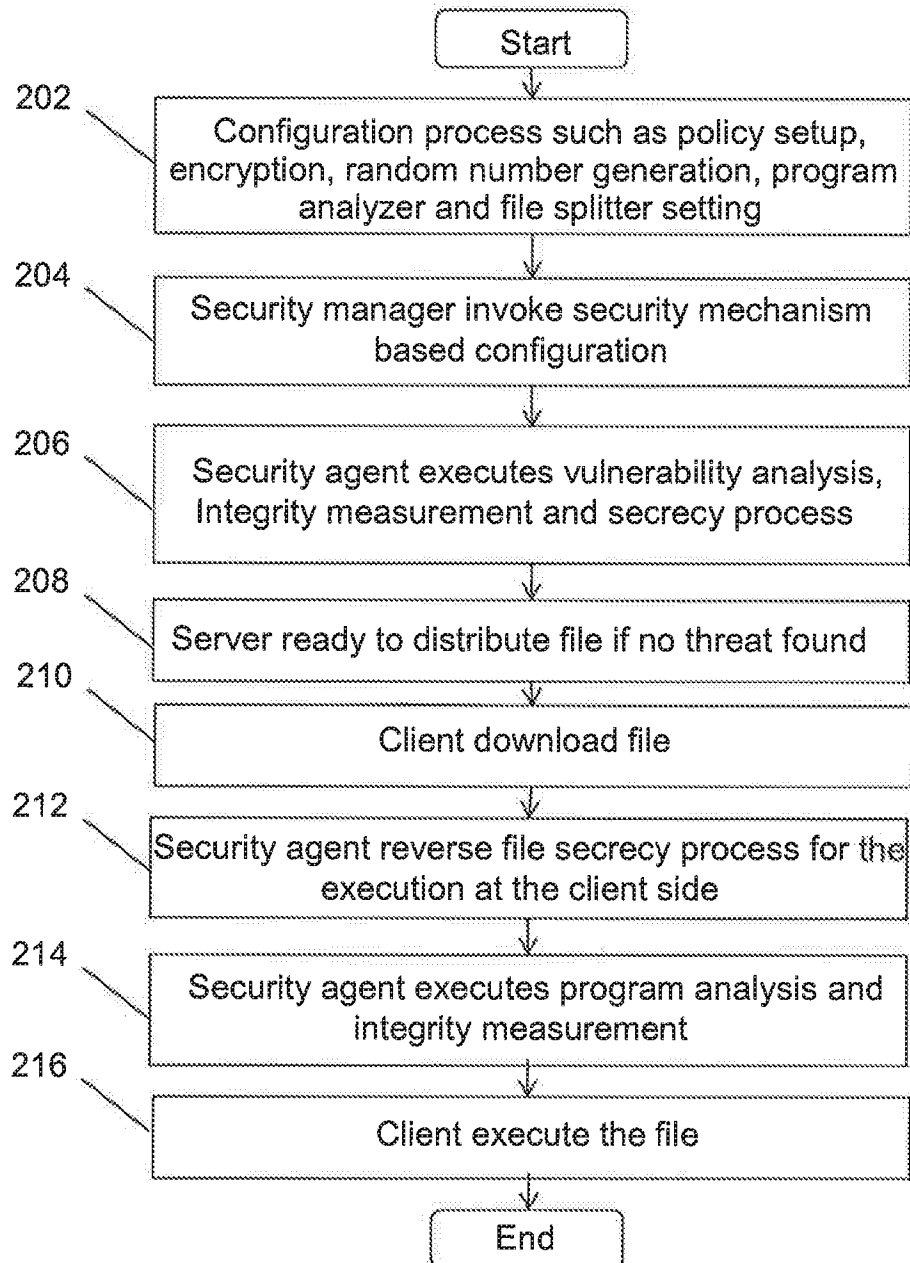


FIG. 2.0

3/8

300

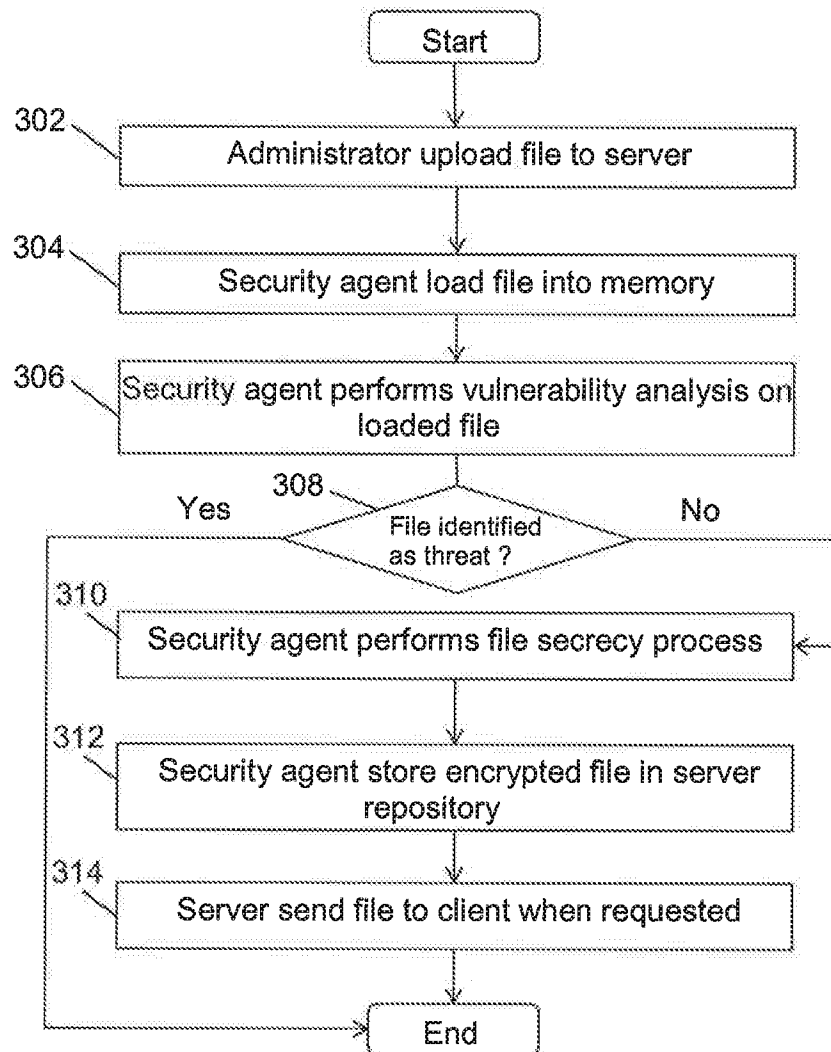


FIG. 3.0

4/8

400

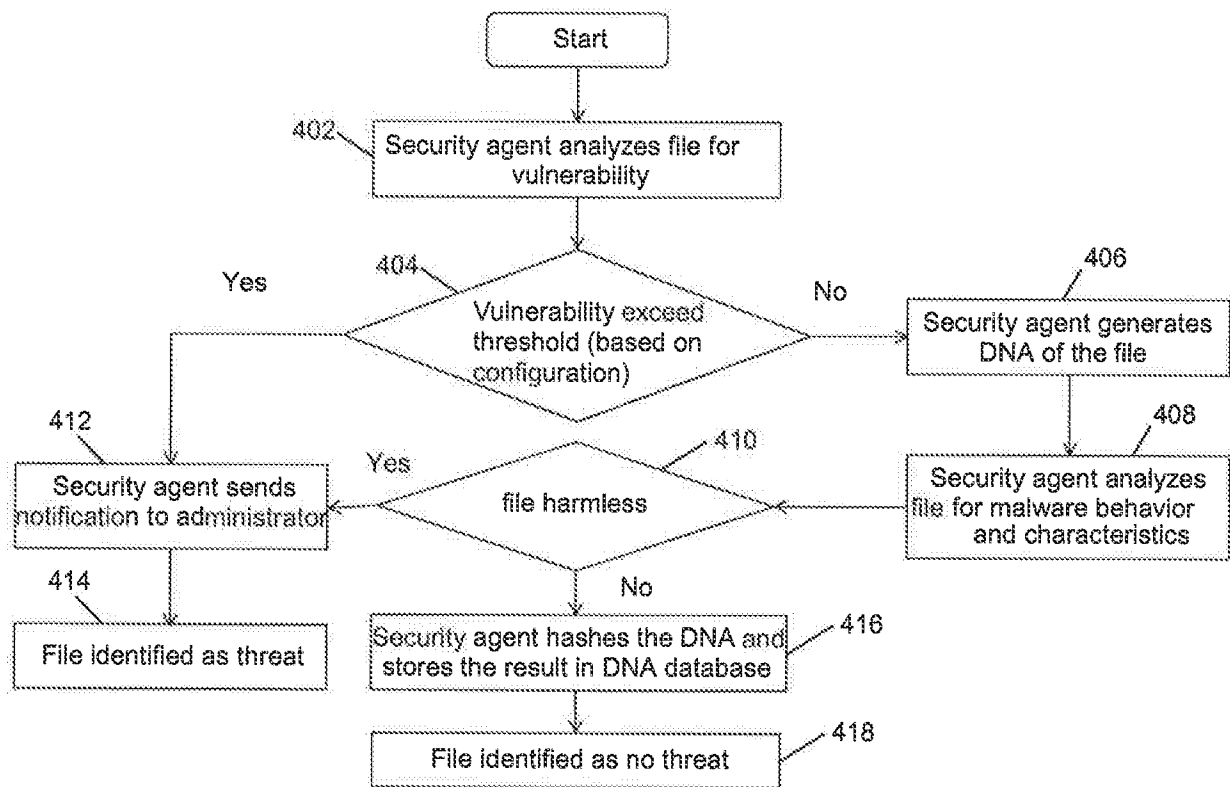


FIG. 4.0

500

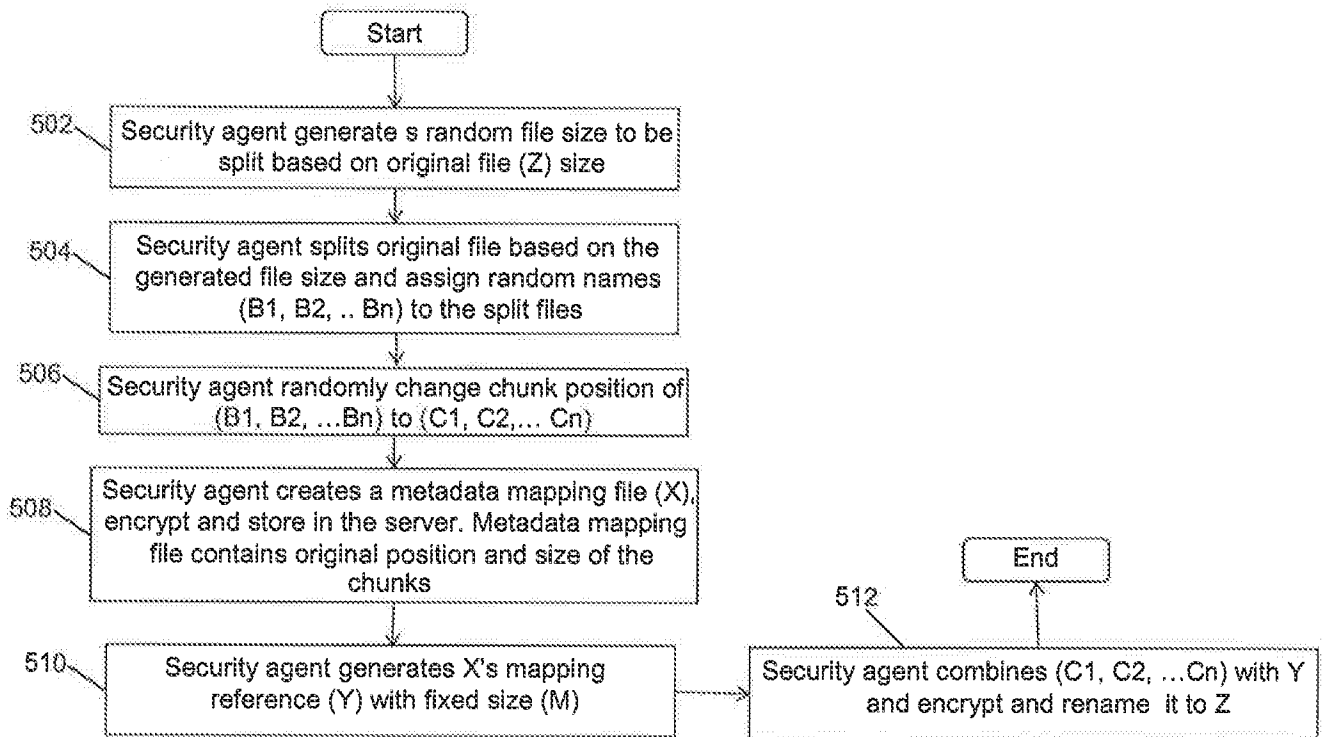


FIG. 5.0

6/8

600

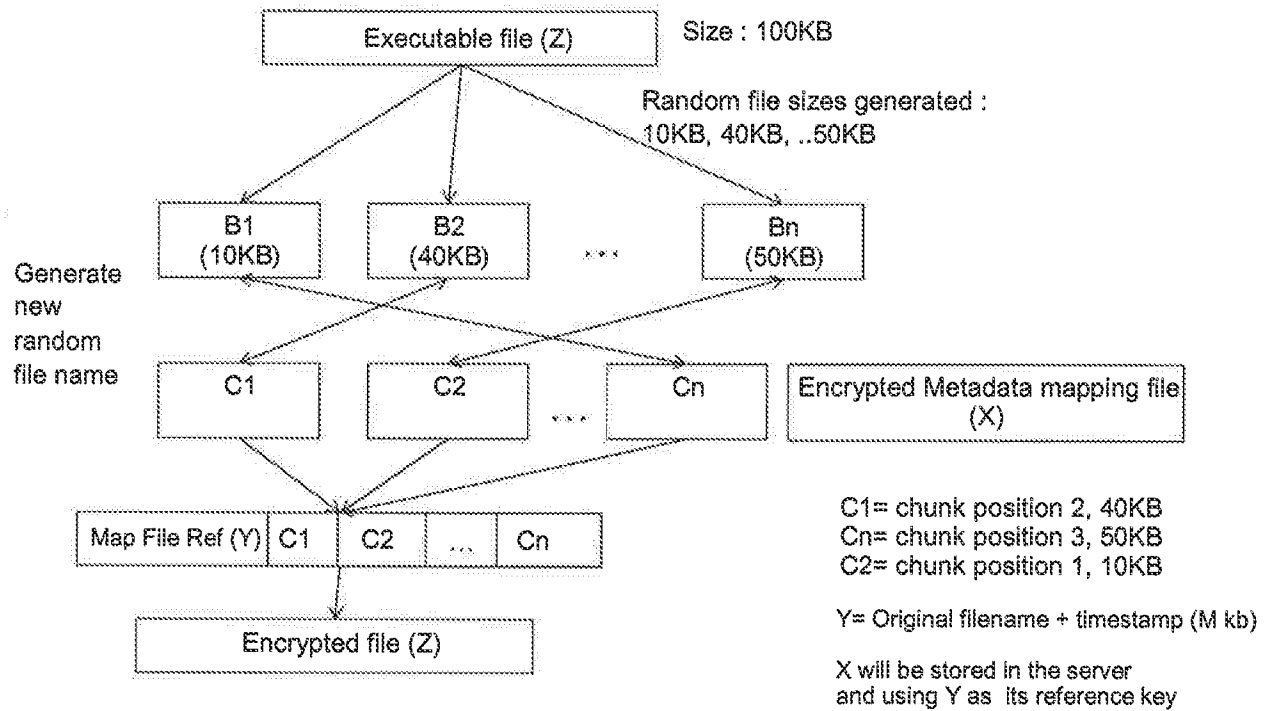


FIG. 6.0

700

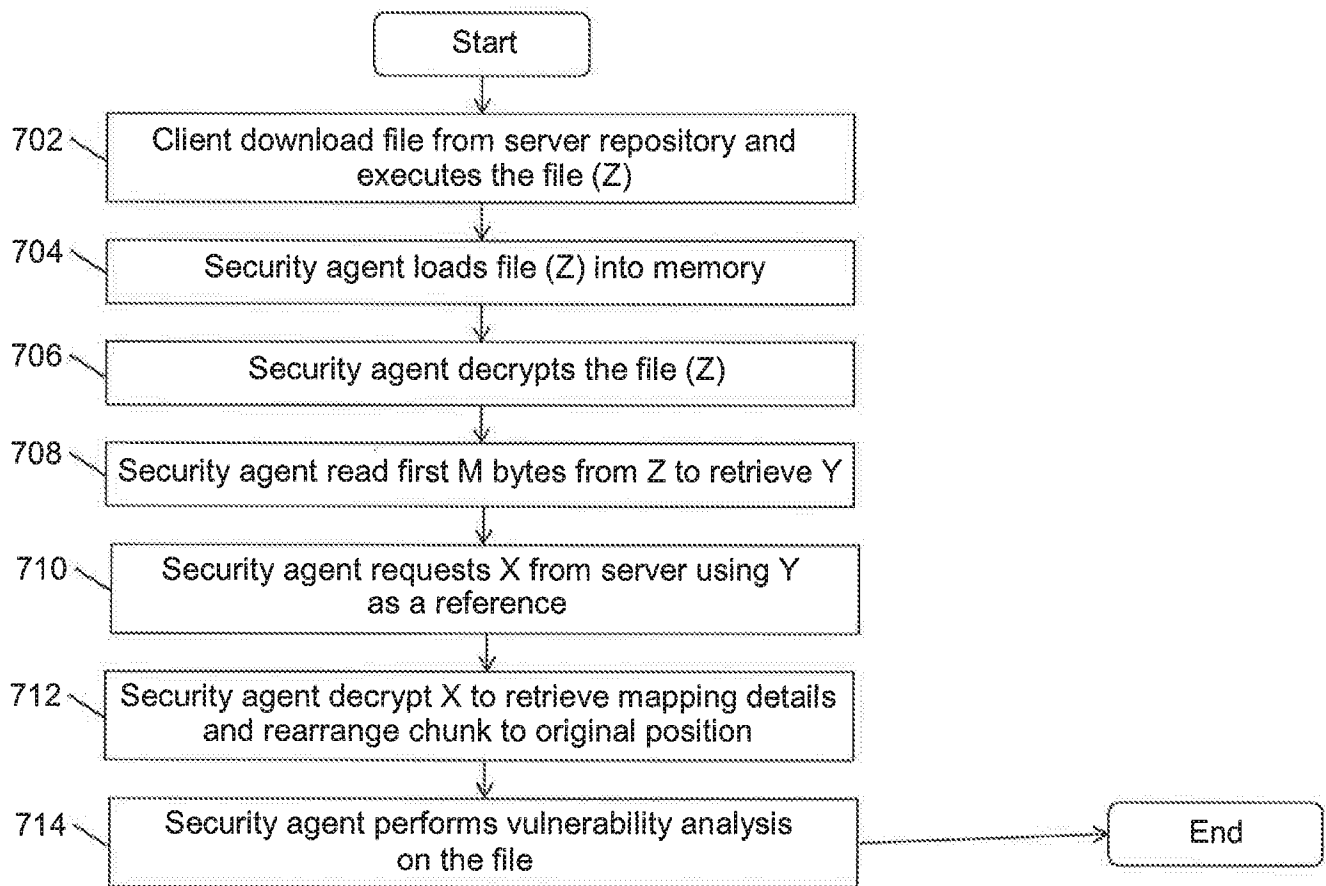


FIG. 7.0

800

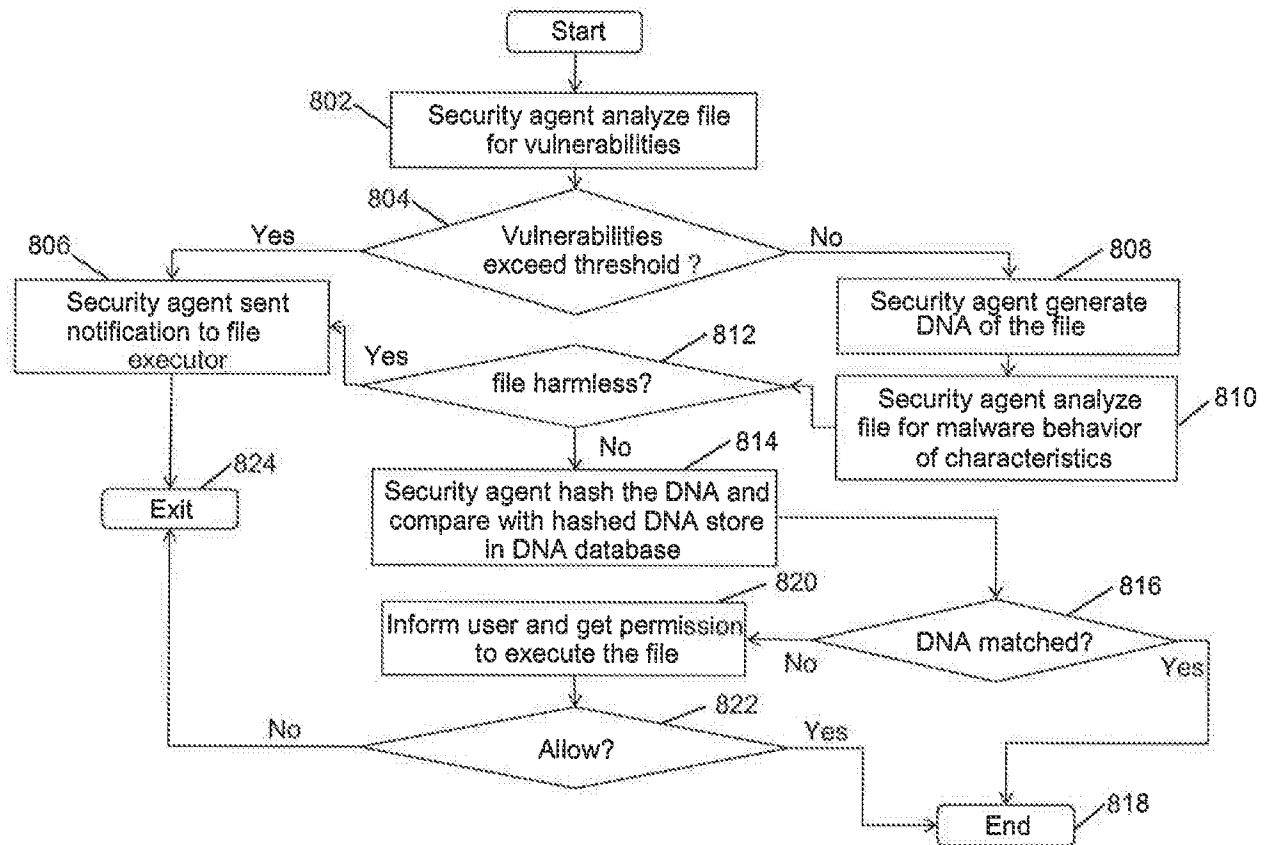


FIG. 8.0

INTERNATIONAL SEARCH REPORT

International application No
PCT/MY2015/050071

A. CLASSIFICATION OF SUBJECT MATTER

INV. G06F21/51 G06F21/60 H04L29/06
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2012/090025 A1 (MILNER STEVE BRADFORD [US] ET AL) 12 April 2012 (2012-04-12)	1,2
Y	paragraph [0016] - paragraph [0045] -----	3-10
Y	EP 1 359 728 A2 (FOLDERMAIL LTD [GB]) 5 November 2003 (2003-11-05) paragraph [0039] - paragraph [0066] -----	3-10



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

12 October 2015

Date of mailing of the international search report

20/10/2015

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Chabot, Pedro

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/MY2015/050071

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2012090025	A1	12-04-2012	NONE

EP 1359728	A2	05-11-2003	EP 1359728 A2 05-11-2003
		US 2004030921 A1	12-02-2004
