



(51) International Patent Classification:
G06F 21/00 (2006.01)

(21) International Application Number:
PCT/US2010/060239

(22) International Filing Date:
14 December 2010 (14.12.2010)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
12/652,851 6 January 2010 (06.01.2010) US

(71) Applicant (for all designated States except US): **ALCATEL-LUCENT USA INC.** [US/US]; 600-700 Mountain Avenue, Murray Hill, NJ 07974-0636 (US).

(72) Inventor; and

(75) Inventor/Applicant (for US only): **RUGGERIO, Raymond, L.** [US/US]; 132 Lincoln Street, Glenview, IL 60025 (US).

(74) Agent: **SANTEMA, Steven, R.**; Alcatel-lucent Usa Inc., Attention: Docket Administrator - Room 3d-201, 600-700 Mountain Avenue, Murray Hill, NJ 07974-0636 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, 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, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, 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.

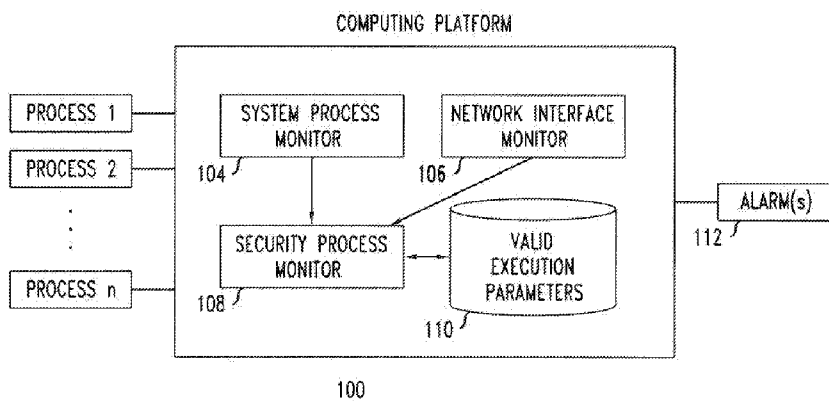
(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, 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, ML, MR, NE, SN, TD, TG).

Published:

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

(54) Title: COMPUTER SECURITY PROCESS MONITOR

FIG. 1



(57) Abstract: A computer security process monitor detects security intrusions of a networked computing platform by monitoring execution statistics associated with one or more computer processes executed by the platform in relation to expected (or "valid") execution parameters. The execution statistics in one example include system process statistics (e.g., process name, peak memory usage, maximum number of threads, peak CPU utilization) and network interface statistics (e.g., IP ports, protocols) associated with the one or more computer processes; and the valid execution parameters define acceptable values or states corresponding to the execution statistics.

805252

COMPUTER SECURITY PROCESS MONITORFIELD OF THE INVENTION

This invention relates generally to computer security and, more particularly, to
5 the security of networked computing platforms within IP-based networks.

BACKGROUND OF THE INVENTION

One of the most important concerns in virtually all computer systems is
security, i.e., the ability to protect information and system resources from intrusions
10 from hackers, malware, viruses, worms or the like. This concern is particularly
worrisome when computing platforms are networked within Internet Protocol (IP)-
based networks that can be accessed by untrusted users/devices and thereby open
windows of vulnerability to the computing platforms.

For example and without limitation, the IP Multimedia Subsystem (IMS) is an
15 architectural framework for delivering converged multimedia services (e.g., voice,
video, data) to end users via an IP platform. IMS components use Session Initiation
Protocol (SIP) for signaling to and from various other IMS components and end users.
SIP messages are ASCII based, easy to read, create and modify; and therefore
vulnerable to eavesdropping, interception and spoofing by untrusted users/devices. A
20 related problem is that IP protocols, IP addressing formats and the like are well
understood and can be spoofed or modified by unscrupulous users/devices.

There are a variety of approaches to address computer security. Anti-virus
software or patches can detect and eliminate threats based on characteristic patterns
(or "signatures") of known threats (e.g., known viruses), or they may employ a set of
25 heuristics or rules to detect threats based on general behaviors. Firewalls protect the
perimeter of a security domain by monitoring, blocking, or proxying communications
from untrusted users that are directed to components inside the firewall. Generally,
however, both of these approaches require periodic updates, reconfiguration or the
like to address constantly evolving numbers and types of threats and may cause some
30 degradation of performance of the protected platform.

805252

SUMMARY OF THE INVENTION

These problems are addressed and a technical advance is achieved in the art by a computer security process monitor, implemented in one example within an embedded IP platform that executes one or more computer processes, whereby security intrusions are detected by monitoring execution statistics associated with the processes in relation to expected (or "valid") execution parameters. The execution statistics in one example include system process statistics (e.g., process name, memory usage, number of threads, CPU utilization) and network interface statistics (e.g., IP ports, protocols) associated with the one or more computer processes; and the valid execution parameters define acceptable values or states corresponding to the execution statistics. Advantageously, the valid execution parameters are derived independent of particular threats or threat characteristics, and hence do not require periodic updates, reconfigurations or the like corresponding to evolving numbers and types of threats.

In one embodiment, there is provided a method comprising maintaining a database including indicia of valid execution parameters associated with one or more computer processes executable by a computing platform; obtaining execution statistics associated with execution of the one or more computer processes by the computing platform; and comparing the execution statistics to the valid execution parameters to detect abnormalities indicative of possible security intrusions.

In another embodiment, there is provided a computing platform comprising a system process monitor, a network interface monitor, a database and a security process monitor. The system process monitor obtains system process information associated with one or more computer processes executed by the computing platform; the network interface monitor obtains network interface information associated with the one or more computer processes; and the database includes indicia of valid execution parameters associated with the one or more computer processes. The security process monitor compares the system process information and the network interface information to the valid execution parameters to detect abnormalities indicative of possible security intrusions.

In yet another embodiment, there is provided a computer security process monitor comprising means for obtaining execution statistics associated with one or

805252

more computer processes executed by a computing platform; means for obtaining indicia of valid execution parameters associated with the one or more computer processes; and means for comparing the execution statistics to the valid execution parameters to detect abnormalities indicative of possible security intrusions.

805252

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other advantages of the invention will become apparent upon reading the following detailed description and upon reference to the drawings in which:

5 FIG. 1 is a block diagram of a computing platform including a computer security process monitor according to an embodiment of the invention;

 FIG. 2 is an exemplary data arrangement of expected (or "valid") execution parameters associated with computer processes executable by a computing platform; and

10 FIG. 3 is a flowchart of an exemplary method for operating a computer security process monitor to detect security intrusions of an embedded IP platform.

DESCRIPTION OF THE PREFERRED EMBODIMENT(S)

 FIG. 1 shows an exemplary computer system 100 comprising a computing
15 platform 102 configured to execute one or more computer processes ("Process 1," "Process 2," Process "n"). In one embodiment, the computing platform 102 comprises an embedded platform that is networked within an Internet Protocol (IP)-based network (for example and without limitation, an IP Multimedia Subsystem (IMS) network); and the processes that are configured to run are known corresponding to
20 known characteristics and functionality of the embedded platform. For example and without limitation, the known characteristics and functionality of the embedded platform may be determined by provisioned parameters such as the particular product/software release and/or its loading parameters (e.g., number of subscribers supported by the platform) or the like. The provisioned parameters are known at time
25 of initialization of the platform and generally do not change throughout the period of operation of the platform, although they are subject to change (albeit infrequently) if/when the platform is re-initialized with product/software updates, reconfigured loading parameters or the like.

 As shown, the computing platform 102 includes a system process monitor 104
30 and network interface monitor 106 for compiling various execution statistics associated with execution of the one or more computer processes. In one example, the execution statistics comprise system process information obtained from the system

805252

process monitor 104 and network interface information obtained from the network interface monitor 106 coincident to execution of the one or more computer processes.

For example and without limitation, the system process monitor 104 may compile system process information including names of one or more processes that are being executed and for each process, the memory usage, number of threads and processor utilization (e.g., CPU utilization); and the network interface monitor 106 may compile network interface information such as indicia of IP ports and protocol being used by the respective processes.

In one embodiment, the system process monitor 104 and network interface monitor 106 are implemented using basic utilities known in the art for compiling execution statistics. For example, the system process monitor 104 may consist of the utility known as "Task Manager" in Microsoft Windows or "System Monitor" in some Linux distributions, which are known to provide a list of processes running in the system, the amount of CPU, memory, and other key resources that are being used; and the network interface monitor 106 may consist of the "Netstat" utility which is known to provide information associated with usage of an IP network interface including IP ports and protocols being used, and a list of active processes communicating over the ports. IP protocols may comprise, for example Transmission Control Protocol (TCP) or User Datagram Protocol (UDP); and IP ports may comprise port numbers (e.g., 0 through 1023) allocated by the Internet Assigned Numbers Authority (IANA).

The computing platform 102 further includes a security process monitor 108 and a valid execution parameter database 110. In one embodiment, the security process monitor 108 obtains execution statistics from the system process monitor 104 and network interface monitor 106 and compares the execution statistics to valid execution parameters obtained from the valid execution parameter database 110 to detect abnormalities that may indicate security intrusions and/or malfunctions associated with execution of the processes. In one embodiment, as a result of the comparison, the security process monitor 108 can detect an invalid process from the execution statistics and/or an invalid parameter that is not consistent with the valid execution parameters and therefore may be indicative of a security intrusion.

For example, consider the case of a malware program that attaches itself to a valid system or application program. In such case, the process name is not changed,

805252

and on the surface a security intrusion is not detected. However, the execution statistics associated with the malware program may reveal abnormalities (i.e., differences from the valid execution parameters) that are indicative of a security intrusion. As will be appreciated, principles of the present invention may also be used to detect abnormalities in valid processes. For example, the detection of an invalid parameter of an otherwise valid process may be indicative of an abnormality such as a "memory leak" (i.e., where the memory used exceeds the expected value) or other malfunction.

The valid execution parameter database 110 includes one or more instances of expected (or "valid") execution parameters corresponding to one or more execution statistics. As will be appreciated, the valid execution parameters may be expressed in a variety of different ways depending on the characteristics of the parameter. For example and without limitation, the valid execution parameters may include text, numerical values, percentages or states; and numerical values and percentages may represent nominal expected values, ranges of values, or minimum or maximum threshold values associated with a particular parameter.

The computer system 100 further includes one or more alarms 112 operably connected to the computing platform to provide alarms/alerts, where appropriate, on condition of the security process monitor detecting an invalid process and/or invalid parameter. Alternatively or additionally, the computer system may record instances of invalid processes or parameters in a log file or the like.

FIG. 2 is an exemplary data arrangement 200 of expected (or "valid") execution parameters associated with computer processes executable by a computing platform. The data arrangement 200 may comprise, for example and without limitation, a list or table of valid execution parameters retrieved from the valid execution parameters database 110. Alternatively or additionally, the data arrangement 200 may include individual instances of valid execution parameters retrieved from the valid execution parameters database 110.

In one embodiment, the data arrangement 200 includes a process name field 202, peak memory usage field 204, maximum threads field 206, peak CPU utilization field 208, and IP port/protocol field 210 and records 220, 230 corresponding to one or more computer processes executable by the computing platform. The process name

805252

field 202 includes the names of the respective processes executable by the computing platform (two shown, a session_mgr process and a fault-mrg process); the peak memory usage field indicates a peak memory usage threshold; the maximum threads field 206 indicates a maximum number of threads; the peak CPU utilization field 208 indicates a peak threshold of CPU utilization (expressed as a percentage of CPU resources); and the IP port/protocol field lists acceptable IP ports/protocols associated with the respective processes.

In particular, the data arrangement 200 indicates for the session_mgr process: 1200 kB peak memory usage, 110 threads (maximum), 45% peak CPU utilization and acceptable IP ports 5060 (for TCP or UDP protocols) or 8000 (for TCP protocol); and for the fault-mrg process: 200 kB peak memory usage, 20 threads (maximum), 20% peak CPU utilization and acceptable IP ports 161 or 162 (UDP protocol).

Turning now to FIG. 3, there is shown an exemplary method for operating a computer security process monitor to detect security intrusions of an embedded IP platform. The steps of FIG. 3 are implemented, where applicable, by the security process monitor 108 in communication with the system process monitor 104, network interface monitor 106 and valid execution parameters database 110.

The method presumes that the valid execution parameters database is populated with one or more valid execution parameters. At step 302, the valid execution parameters database is "initialized" or populated with one or more instances of valid execution parameters corresponding to characterized software/processes that the computing platform will run. Once initialized, it is contemplated that periodic updates to the valid execution parameters (e.g., corresponding to a new product/software release) may also be required to update one or more parameters and step 302 is intended to encompass both initial initialization and "re-initialization" following periodic updates. In one embodiment, following the initialization/update, the valid execution parameters database is locked with vendor specific passwords/keys, such that modifications to the parameters can not be made by users not having the passwords/keys.

In one embodiment, the security process monitor begins its operation by retrieving execution statistics, including system process information and network interface information, from the system process monitor 104 and network interface

805252

monitor 106 respectively. System process information is retrieved from the system process monitor at step 304 and network interface information is retrieved from the network interface monitor at step 306. Alternatively or additionally, execution statistics may be obtained independently by the security process monitor or may be
5 obtained from sources other than the system process monitor or network interface monitor.

At step 308, the security process monitor compares the execution statistics to the corresponding valid execution parameters obtained from the valid execution parameters database to detect abnormalities (i.e., differences from the valid execution
10 parameters) that may be indicative of a possible security intrusion or malfunction of a valid process. For example and without limitation, the execution statistics may indicate a process name (either explicitly or via derivation from the execution statistics) that is not registered in the list of valid processes from the valid execution parameters database, thereby defining an invalid process. The execution statistics
15 may further reveal one or more invalid parameters other than the process name that are inconsistent with the valid execution parameters.

If an invalid process (i.e., process name) is detected, determined at step 310, the security process monitor activates an alarm and/or generates a record of the event at step 314 to provide indicia of the invalid process to an operator/user. The indicia of
20 the invalid process may include, for example and without limitation, the process name(s) and/or parameters that are identified or suspected to be invalid by the security process monitor.

If an invalid process is not detected but an invalid parameter is detected, determined at step 312, the security process monitor activates an alarm or generates a
25 record at step 314 to provide indicia of the invalid parameter(s) to an operator/user. The indicia of invalid parameter(s) may include, for example and without limitation, indicia of excessive memory usage (indicative of a "memory leak") associated with an otherwise valid process.

Until such time as there are no further comparisons to be made, determined at
30 step 316, the process returns to step 308 to continue evaluating the execution statistics against the valid execution parameters. As will be appreciated, the evaluation may be performed continuously or at periodic intervals depending on the implementation.

805252

FIGS. 1-3 and the foregoing description depict specific exemplary embodiments of the invention to teach those skilled in the art how to make and use the invention. The described embodiments are to be considered in all respects only as illustrative and not restrictive. The present invention may be embodied in other
5 specific forms without departing from the scope of the invention which is indicated by the appended claims. All changes that come within the meaning and range of equivalency of the claims are to be embraced within their scope.

For example, the term "computing platform" as used herein is generally defined as any platform or combination of platforms, including but not limited to
10 embedded IP platforms, that is operable to execute one or more computer processes and that is subject to potential security intrusions. As is well understood in the art, a computing platform includes respective processors and memory (not shown) for executing programming instructions residing in software code to effect certain transactions.

15 The term "computer process" as used herein is generally defined as an execution of instructions by the computing platform to perform a particular task or application program, or portion thereof. Each application has at least one process, most typically having a process name that corresponds to the application (e.g., Microsoft Outlook has a process named outlook.exe). Each process may consist of
20 one or more "threads" defining separate parts of the process which may be run concurrently.

805252

WHAT IS CLAIMED IS:

1. A method comprising:
 - maintaining a database including indicia of valid execution parameters associated with one or more computer processes executable by a computing platform;
 - 5 obtaining execution statistics associated with execution of the one or more computer processes by the computing platform;
 - comparing the execution statistics to the valid execution parameters to detect abnormalities indicative of possible security intrusions.
- 10 2. The method of claim 1, wherein the step of obtaining execution statistics comprises obtaining system process information associated with the computing platform.
- 15 3. The method of claim 2, wherein the step of obtaining system process information comprises obtaining one or more of: process name, memory usage, number of threads, and CPU utilization associated with one or more computer processes executed by the computing platform.
- 20 4. The method of claim 1, wherein the step of obtaining execution statistics comprises obtaining network interface information associated with the computing platform.
- 25 5. The method of claim 4, wherein the step of obtaining network interface information comprises obtaining one or more of: IP port information and indicia of IP protocol associated with one or more computer processes executed by the computing platform.
- 30 6. The method of claim 1, wherein the step of obtaining execution statistics comprises obtaining system process information and network interface information associated with one or more computer processes executed by the computing platform.

805252

7. The method of claim 6, wherein

the step of obtaining system process information comprises obtaining one or more of: process name, memory usage, number of threads, and CPU utilization associated with execution of the one or more computer processes; and

5 the step of obtaining network interface information comprises obtaining one or more of: IP port information and indicia of IP protocol associated with execution of the one or more computer processes.

8. The method of claim 1, wherein the step of comparing comprises:

10 detecting an invalid process executed by the computing platform, the invalid process defining a process identifiable from the execution statistics that is not consistent with the valid execution parameters.

9. The method of claim 1, wherein the step of comparing comprises:

15 detecting an invalid parameter executed by the computing platform, the invalid parameter defining a parameter identifiable from the execution statistics that is not consistent with the valid execution parameters.

10. A computing platform comprising:

20 a system process monitor operable to obtain system process information associated with one or more computer processes executed by the computing platform;

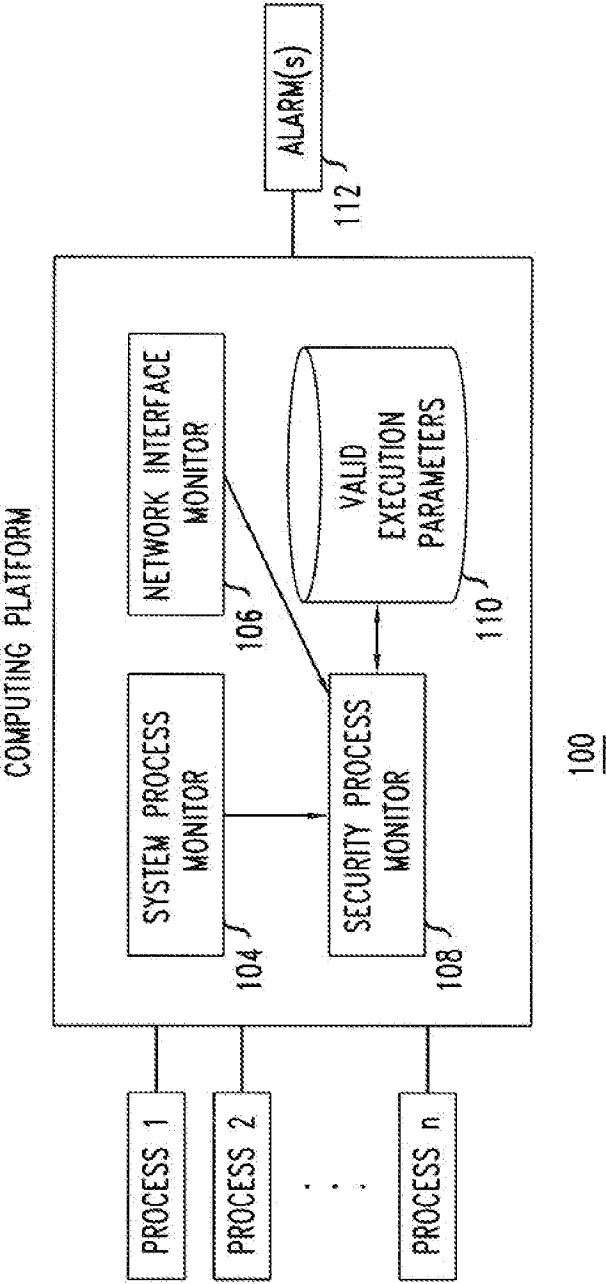
a network interface monitor operable to obtain network interface information associated with the one or more computer processes;

25 a database including indicia of valid execution parameters associated with the one or more computer processes; and

a security process monitor operably coupled to the system process monitor, network interface monitor and database, that is operable to compare the system process information and the network interface information to the valid execution parameters to detect abnormalities indicative of possible security intrusions.

30

FIG. 1



2/3

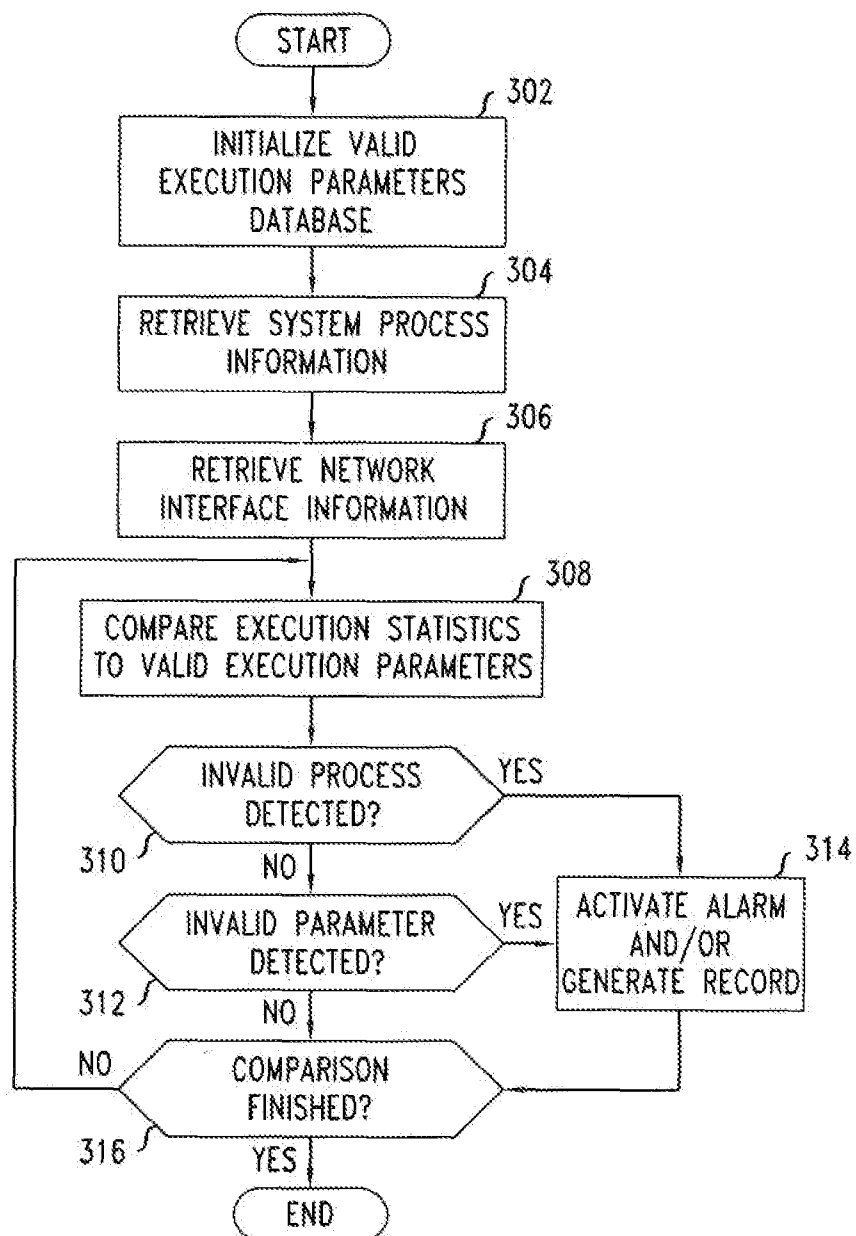
FIG. 2

200

	202	204	206	208	210
	PROCESS NAME	PEAK MEMORY USAGE	MAX# THREADS	PEAK CPU UTILIZATION	IP PORT USAGE/PROTOCOL
220	session_mgr	1200 kB	110	45%	5060-TCP 5060-UDP 8000-TCP
230	fault_mgr	200 kB	20	20%	161-UDP 162-UDP

3/3

FIG. 3



INTERNATIONAL SEARCH REPORT

International application No

PCT/US2010/060239

A. CLASSIFICATION OF SUBJECT MATTER

INV. G06F21/00

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

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 practical, search terms used)

EP0-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2009/097610 A1 (UNIV NORTHEASTERN [US]; MOFFIE MICHA [US]; KAEI DAVID [US]; COHEN AVI) 6 August 2009 (2009-08-06) paragraph [0007] paragraph [0013] - paragraph [0014] paragraph [0039] - paragraph [0131] -----	1-10
X	US 7 089 428 B2 (FARLEY TIMOTHY P [US] ET AL) 8 August 2006 (2006-08-08) column 3, line 25 - column 5, line 16 column 6, line 16 - column 34, line 46 -----	1-10
A	WO 02/45380 A2 (LANCOPE INC [US]; COPELAND JOHN A III [US]) 6 June 2002 (2002-06-06) the whole document -----	1-10

☐

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 document 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

25 February 2011

Date of mailing of the international search report

04/03/2011

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

Pinto, Raúl

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2010/060239

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
W0 2009097610 A1	06-08-2009	US 2011004935 A1	06-01-2011
US 7089428 B2	08-08-2006	AU 6295801 A	12-11-2001
		EP 1277326 A2	22-01-2003
		JP 2004537075 T	09-12-2004
		W0 0184285 A2	08-11-2001
		US 2002078381 A1	20-06-2002
W0 0245380 A2	06-06-2002	AT 344573 T	15-11-2006
		AU 3054102 A	11-06-2002
		CA 2430571 A1	06-06-2002
		DE 60124295 T2	31-05-2007
		EP 1338130 A2	27-08-2003