(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
17 January 2013 (17.01.2013)

WIPO | PCT

(10) International Publication Number
WO 2013/009302 A1

(51) International Patent Classification:
G06F 21/22 (2006.01) *G06F 9/06* (2006.01)

(21) International Application Number:
PCT/US20 11/043716

(22) International Filing Date:
12 July 201 1 (12.07.201 1)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant *(for all designated States except US):* **HEW¬ LETT-PACKARD DEVELOPMENT COMPANY , L.P.** [US/US]; 11445 Compaq Center Drive W, Houston, Texas 77070 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only):* **JEANSONNE, Jeff** [US/US]; 11445 Compaq Center Drive W, Houston, Texas 77070 (US). **ALI, Vali** [US/US]; 11445 Compaq Center Drive W, Houston, Texas 77070 (US). **MANN, James M.** [US/US]; 11445 Compaq Center Drive W, Houston, Texas 77070 (US).

(74) Agents: **HABLINSKI, Reed J.** et al; Hewlett-Packard Company, Intellectual Property Company, 3404 East Harmony Road, Mail Stop 35, Fort Collins, Colorado 80528 (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,

[Continued on next page]

(54) Title: COMPUTING DEVICE INCLUDING A PORT AND A GUEST DOMAIN

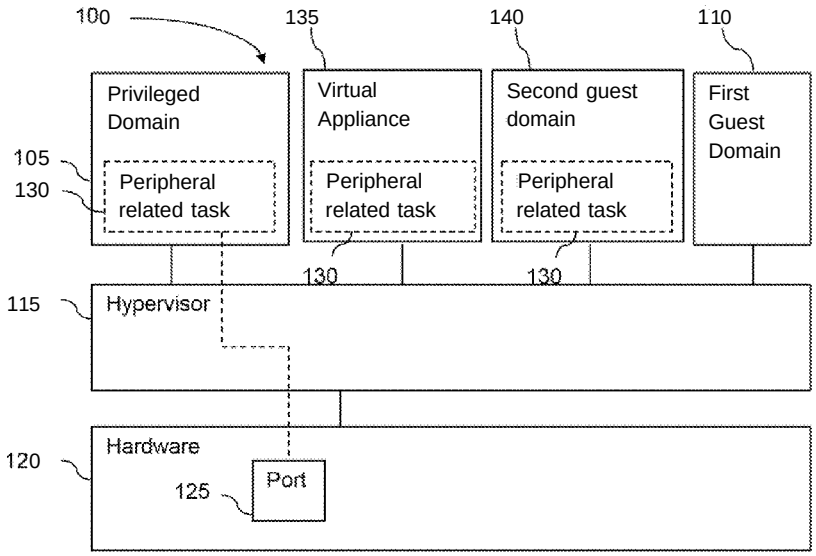

FIG. 1

[Continued on next page]

WO 2013/009302 A1

TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to the identity of the inventor (Rule 4.17(i))*

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

Published-

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

(57) Abstract: A first guest domain and an isolated peripheral related task. A peripheral related task to communicate with the peripheral and prevent the first guest domain from communicating with the peripheral.

Computing device including a Port and a Guest domain

Background

[0001] A virtual machine is a software implementation of a machine that executes instructions like a physical machine. **The** virtual machine is susceptible to the same malicious attacks as a physical machine. Plug and play can allow a peripheral to be connected to a machine without user involvement to configure the peripheral. If the peripheral was malicious it may infect the virtual machine.

Brief Description Of The Prawings

[0002] Some embodiments of the invention are described with respect to the following figures:

Fig. 1 is a block diagram of a computing device according to an example implementation;

Fig. 2 is a block diagram of a computing device according to an example **implementation;**

Fig. 3 is an interface according to an example **implementation;**

Fig. 4 is a flow diagram **of** a method of communicating with a peripheral connected to a computing device according to an example implementation;

Fig. 5 is a flow diagram of a method of communicating with a peripheral connected to a computing device according to an example implementation; and

Fig. 6 is a computing system including a computer readable medium according to an example implementation.

Detailed Description

[0003] A computing device may be susceptible to attacks by malicious code. The computing device may be for example a server, desktop, notebook, cell phone, PDA, or another computing device. **The** malicious code may be for example, **malware,** viruses, firmware attacks or other. A computing device can execute an operating system which may be attacked by viruses or malware.

[0004] A virtual machine may also be known as a virtual domain for hosting an operating system executing in the virtual domain. A physical machine may execute multiple domains. An operating system executing on a domain is susceptible to an attack by viruses and matware that can attack the operating system if executing directly on the physical hardware of the computing device. The domains can be managed and isolated from one another by a hypervisor also known as a virtual machine monitor or in conjunction with one of the domains such as a privileged domain. Each domain on a computing device can execute a peripheral related task. A peripheral related task can be logic or instructions that determine if a peripheral is malicious, A virtual appliance may be used for the purpose of executing the peripheral related task. A virtual appliance can run in a domain. The peripheral related task can scan a peripheral that is attached to the computing device to prevent the peripheral from attacking another domain.

[0005] In one example a computing device includes a first guest domain and a peripherai related task isolated from the first guest domain. A port can connect the computing device to a peripheral device. A privileged domain can cause the peripheral related task to be executed to identify the peripheral device. The peripheral related task prevents the first guest domain from communicating with the peripherai if it is determined malicious.

[0006] In another example, a method of communicating with a peripheral connected to a computing device includes isolating a peripheral related task from a first guest domain. A virtual appliance can be generated to execute the peripheral related task. The virtual appliance can receive from the peripheral an indication of the type of peripheral. The virtual appliance can validate the type of peripheral. Communication with the peripheral by the first guest domain can be prevented until the type of peripheral is validated.

[0007] With reference to the figures, Fig. 1 is a block diagram of a computing device 100 according to an example implementation. The computing device 100 can include a first guest domain 110. A privileged domain 105 can be separate from the first guest domain 110. In one example, the privileged domain 105 may not be

a!lowed to execute the peripherai **related** task 130 and may generate a virtual appliance or another guest domain to execute the peripherai related task 130 to prevent the privileged domain 105 from being attacked by a malicious peripheral. The privileged domain 105 may not allow changes to **be** made to the privileged domain 105 by other domains that may connect to a potentially malicious peripheral such as a second guest domain 140, or a virtual appliance 135. In another example the privileged domain 105 may be allowed to execute the peripheral related task 130 if the privileged domain 105 is not susceptible to an attack from a malicious peripheral.

[0008] The hardware 120 can include a port 125 to connect a peripheral device. A hypervisor 115 can manage the hardware 120 resources. A peripheral related task 130 can **be** executed in a domain other than the first guest domain for example the privileged domain 105, a virtual appliance 135 or second guest domain 140. **The** peripheral related task 130 may be instructions to identify the peripheral device. The peripheral related task 130 can prevent the first guest domain 110 from accessing the peripheral if it is determined malicious.

[0009] The port 125 is an interface through which data is transferred between a computer and another device. The port can foe for example a wired port such as a universal serial bus (USB) port, an IEEE 1394 port, a thunderbolt port, a sata port or another wired connection. The port 125 may be a wireless port such as a Bluetooth® port, a wifi port, a **wwan** port or another wireless connection. The other device can be a peripheral, for example, a printer, mouse, keyboard, monitor, a storage device, network device or another peripheral

[0010] The hypervisor 115 is a layer for initially communicating directly with hardware 120 replacing the operating system to allow the hardware to run multiple guest operating systems concurrently within multiple domains. In some implementations the hypervisor 115 initiates a domain, such as privileged domain and maps the input/output (I/O) controller to privileged domain to communicate directly with the hardware 120 rather than the hypervisor. In one embodiment, a computer executing a hypervisor may contain three components. The first

component is the hypervisor 115 and the second component is the privileged domain 105 which may also be known as domain 0 (DomO). The privileged domain can be a privileged guest running on the hypervisor 115 with direct hardware access and guest management responsibilities. The third component is a Domain U which can be an unprivileged domain guest (DomU). The DomU can be an Unprivileged guest running on the hypervisor which has no direct access to hardware such as the memory, hard disk, a port or any other hardware 120. The first guest domain can be an example of a DomU.

[0011] The peripheral related task 130 can be an application that is executed by a domain. If a peripheral is connected to the port the peripheral may send an indication of what type of device the peripheral is. For example the peripheral may indicate that it is a storage device which may cause the execution of the peripheral related task 130. The execution may be on any of the domains other than the first guest domain such as another guest domain, a virtual appliance, the privileged domain or the hypervisor. For example the peripheral related task may challenge the peripheral by trying to store and retrieve information from the storage device. The task may also scan for malicious content. A peripheral may be malicious when it includes for example a virus, malware or another destructive program that takes advantage of security hole in a domain. A privileged domain is intended to be un¬ susceptible to viruses and malware, this can be because for example the privileged domain includes trusted software and may not allow writing to the domain by another domain. The privileged domain can execute the peripheral related task 130 for the peripheral device which may cause a malicious code to infect an unprivileged domain but not the privileged domain. Once the peripheral related task 130 has verified that the peripheral is not malicious then an unprivileged domain such as the first guest domain 110 may access the peripheral device. There may be multiple different levels of access that can be given to the first guest domain 110.

[0012] Fig. 2 is a block diagram of a computing device according to an example implementation. The computing device 200 can include, hardware 220 which can include a port such as a wired port 225 or a wireless port 245. The wired port 225 can be for example a universal serial port, an IEEE 1394 port, a thunderbolt port, a

sata port or another wired connection. The wireless port 245 can be a port such as a Bluetooth port, a wifi port, a wwan port or another wireless connection.

[0013] A domain such as a privileged domain 205 that is outside of and **isolated** from the first guest domain 210 as a secure quarantine area for ail peripheral devices where they can initially be enumerated, analyzed, authenticated, and /or remediated as necessary before being exposed to a user operating environment. In addition, some types of devices may be blocked entirely from the first guest domain 210.

[0014] Once the hypervisor maps the s/o controller to the privileged domain then the privileged domain is the domain that first enumerates any peripheral that is presented at the hardware 220 level to a port controller. Policy decisions can be made at this level, but the privileged domain 205 can be a highly secure environment. Because of the highly secure environment the peripheral can be connected to a virtual appliance 235 whose sole purpose is to enforce policy settings related to the peripheral device. This virtual appliance 235 can make a decision on how to, or even whether to, expose the peripheral device to a first guest domain 210 based on pre-configured policy settings related to a number of possibie mechanisms, device class authentication, device class configuration policy enforcement, device class white list or black list, specific device white list, abstracted user interaction, device class authentication or another policy setting.

[0015] As an example, for USB human interface devices, the privileged domain detects device insertion. Subsequent exposure of said human interface device to first guest domain 210 is delayed until the device is analyzed. The privileged domain treats the device as hostile until it can be authenticated by the peripheral related task as being a device as indicated. For human interface devices, such as a keyboard or mouse, this could be performed by presenting a challenge to the user via the display subsystem. This may be done through the secure graphical user interface so that the challenge is not visible to any guest domains such as the first guest domain 210. The challenge may include presenting random characters to the user as well as a graphical keyboard and waiting for a user to enter the characters by either clicking

the correct sequence of buttons on the graphical keypad with the mouse or fay entering the characters using the keyboard. The peripheral related task 230 can assure that the resulting input is coming from the device that was inserted. In this way, the peripheral related task can authenticate that the device is indeed acting as a human interface device for the machine operator and not simply posing as a human interface device.

[0016] A hypervisor 215 can manage the domains such as the privileged domain 205 and the first guest domain 210. In managing the domains the hypervisor can give the privileged domain access to the wired port 225 or the wireless port 245. This can prevent the first guest domain 210 from accessing a peripheral connected to the wired port 225 or the wireless port 245. In one embodiment the privileged domain has direct access to the hardware 220 and the first guest domain 210 does not have direct access to the hardware 220.

[0017] The first guest domain 210 may include an interface that can be used to determine the amount of access the first guest domain 210 has to a peripheral connected to a wired port 225 or a wireless port 245. The first guest domain may have for example full access to the peripheral, may have no access to the peripheral, may receive information about the peripheral in text so that it is sure that malicious instructions are not embedded in the data transfer and received by the first guest domain. The text may be in an ascii format and may be a list of files on the peripheral device if the peripheral device is a storage device. The privileged domain 205 or the peripheral related task may create the list of text representing the files on the peripheral device. A user may then be able to select a file that would be accessible to the first guest domain while others would continue to be identified by a text representation. If a file was selected then the privileged domain 205 could send the file to the first guest domain 210 or could allow the first guest domain 210 to access the peripheral device through the privileged domain 205. This could be done through simple remote procedure call (RFC) or other intra-domain communication mechanisms in which only text information of the files is transferred (filenames, sizes, r/w/x attributes, modification dates, etc). A dialog could be presented to the user allowing them to either allow the peripheral for full insertion' into the file system

of the first guest domain 210, rejection of the peripheral, or something in between. An example of "something in between" is the user could decide to transfer to/from the storage device over a communication channel such as text over RPC, rather than allowing it to be inserted into the first guest domain environment as part of the first guest domain's file system.

[0018] The peripheral related task may also include logic to determine if an auto run file is on the peripheral device. The logic may be in a privileged domain, virtual appliance or another guest domain and can prevent the first guest domain 210 from accessing the auto run file. An auto run file is a file that a domain may search for when a peripheral device is connected. If an auto run file is detected then the domain may run the application or instructions in the auto run file. If the auto run file was to install malicious software a user may install the malicious software by connecting a peripheral to a port on the computing device 200, however by the peripheral related task 240 removing the auto run file or preventing the first guest domain 210 from accessing the auto run file the first guest domain may not automatically install malicious software from: a peripheral device.

[0019] The peripheral related task 230 may include or have access to a blacklist 250. The black list 250 may include a list of peripheral devices that the first guest domain 210 is prevented from accessing. The peripheral related task 230 may also have access to a white list which is a list of devices that the system may be able to access without performing task on prior to allowing the first guest domain 210 access to the peripheral.

[0020] The privileged domain 205 black list 250 policy can be configured such that all of a certain type of device is blocked from being exposed to the first guest domain 210. For example, a policy may be set to instruct the privileged domain 205 to block alt USB mass storage class devices from being exposed to the first guest domain 210.

[0021] This policy may include a "learn mode" which can enable an administrator to connect a known good device to a platform, at which time the privileged domain 205 can store the device information for later comparison. In normal operation,

whenever a peripheral device was attached to a port, the privileged domain can compare each device to the white list and require a match before passing it to the first guest domain 210. This could be very restrictive in that it can only allow devices with the information such as a serial number already in the white fist such that the particular device in the white list worked, or it could be configured to be less restrictive such that the serial number were ignored and all those particular devices can be passed through the first guest domain 210.

[0022] The peripheral related task 230 may be able to execute a scanner 240. The scanner 240 can scan the contents of the peripheral device for malicious code prior to allowing access to the peripheral device by the first guest domain 210. For example the scanner may scan the contents of the peripheral device for viruses, matware, or other malicious code. The scanner may be able to remove the viruses from the peripheral prior to giving the first guest domain 210 access to the peripheral device or may allow the first guest domain 210 to access materials that were scanned and shown to be free of a virus or malware.

[0023] Fig. 3 is an interface according to an example implementation. The interface 300 may be a secure graphical user interface. The interface can be used to select the level of access the first guest domain has to communicate with the peripheral device. For example the interface may ask the user to select the level of access for a peripheral device that has been detected by the privileged domain. Examples of the options may be to reject the device, integrate the device as part of the file system, or communicate with the peripheral device over a secure channel. The interface may allow a user to create or manage a policy that is implemented by the peripheral related task or the privileged domain such as creating a white list or black list.

[0024] Fig. 4 is a flow diagram of a method of communicating with a peripheral connected to a computing device according to an example implementation. The method includes isolating a peripheral related task from a first guest domain 210 at 405. The peripheral related task that is isolated from the first guest domain may be a peripheral related task 130. A virtual appliance can be generated at 410 to execute

the peripheral related task 130. The generation of the virtual appliance can be initiated by the privileged domain. The virtual appliance can receive from the peripheral an indication of the type of peripheral at 415. The type of peripheral may be for example a storage device, a human interface device such as a keyboard or mouse, or an output device such as a display or printer. The virtual appliance can validate the type of peripheral at 420. The validation may include asking the user to type a random code on the keyboard this can prevent a storage device from identifying itself as a keyboard and causing keyboard input such as starting programs. The communication with the peripheral by the first guest domain can be prevented at 425 until the type of peripheral is validated.

[0025] Fig. 5 is a flow diagram of a method of communicating with a peripheral connected to a computing device according to an example implementation.. The method includes isolating a peripheral related task from a first guest domain 210 at 505. The peripheral related task that is isolated from the first guest domain may be a peripheral related task 130. A peripheral related task 130 can be executed at 510 by a virtual appliance. The virtual appliance can perform tasks, such as the peripheral related tasks that may not be executed by a privileged domain. The virtual appliance can receive from the peripheral an indication of the type of peripheral at 515. The type of peripheral may be for example a storage device, a human interface device such as a keyboard or mouse, or an output device such as a display or printer. The virtual appliance can validate the type of peripheral at 520. The validation may include asking the user to type a random code on the keyboard. This can prevent a storage device from identifying itself as a keyboard and causing keyboard input such as starting programs. The communication with the peripheral by the first guest domain 110 can be prevented at 525 until the type of peripheral is validated.

[0026] The method can include scanning the contents of the peripheral device for malicious code prior to allowing access to the peripheral device by the first guest domain at 530. The scan of the malicious code may include a virus scan, malware scan or another scan. The level of access the first guest domain has to communicate with the peripheral device can be selected at 535. The level of access

can be based on policies implemented by the peripheral related task. The policies may be predetermined or may be selected by the user of the first guest domain. The method may include determining if an auto run file is on the peripheral device at 540. The peripheral related task can prevent the first guest domain from accessing the auto run file. The peripheral related task may remove the auto run file, prevent access to the auto run file by the first guest domain, or allow only secure communications with the files on the peripheral device such as only showing an ascii text based listing of the files on the periphera! device.

[0027] Fig. 6 is a computing system including a computer readable medium according to an example implementation. The non-transitory computer readable 615 or 616 medium can include code such as a domain or a peripheral related task that can be executed by a processor 605. The processor 605 can be connected to a controller hub 610. The controller hub can connect to the display 630 through a graphics controller 620, a keyboard 635, a mouse 640 and a sensor 645 such as a webcam. The keyboard 635, mouse 640, display 630, sensor 645 and computer readable media 615 and 616 are some examples of peripherals devices that can be connected to the computing device 600 through a port. The controller hub may include the port or there may be other components between the peripheral and the controlier hub 610 that allows communication between the peripheral and the processor 605.

[0028] The privileged domain if executed can cause a computing device to isolate a periphera! related task from a first guest domain. The privileged domain can cause a virtual appliance to be generated to execute the peripheral related task. The virtual appliance can receive from the peripheral an indication of the type of periphera!. The periphera! related task can validate the type of periphera! and prevent communication with the peripheral by the first guest domain until the type of periphera! is validated. The peripheral related task may scan the contents of the periphera! device for malicious code prior to allowing access to the periphera! device by the first guest domain. The periphera! related task may allow the selection, through an interface, of the level of access the first guest domain has to communicate with the periphera! device.

[0029] The techniques described above may be embodied in a computer-readable medium for configuring a computing system to execute the method. The computer readable media may include, for example and without limitation, any number of the following: magnetic storage media including disk and tape storage media; optical storage media such as compact disk media (e.g., CD-ROM, CD-R, etc.) and digital video disk storage media; holographic memory: nonvolatile memory storage media including semiconductor-based memory units such as FLASH memory, EEPROM, EPROM, ROM; ferromagnetic digital memories; volatile storage media including registers, buffers or caches, main memory, RAM, etc.; and the Internet, just to name a few. Other new and various types of computer-readable media may be used to store the software modules discussed herein. Computing systems may be found in many forms including but not limited to mainframes, minicomputers, servers, workstations, personal computers, notepads, personal digital assistants, various wireless devices and embedded systems, just to name a few.

[0030] In the foregoing description, numerous details are set forth to provide an understanding of the present invention. However, it will be understood by those skilled in the art that the present invention may be practiced without these details. While the invention has been disclosed with respect to a limited number of embodiments, those skilled in the art will appreciate numerous modifications and variations therefrom. It is intended that the appended claims cover such modifications and variations as fall within the true spirit and scope of the invention.

What is claimed is:

1. A computing device comprising:
a first guest domain;
a peripheral related task isolated from the first guest domain;
a port to connect a peripheral device; and
a privileged domain to cause the peripheral related task to be executed to identify the peripheral device, wherein the peripheral related task prevents the first guest domain from communicating with the peripheral if it is determined malicious.

2. The device of claim 1, wherein the privileged domain generates a virtual appliance to execute the peripheral related task.

3. The device of claim 1, wherein the port is a wired or a wireless port.

4. The device of claim 1, further comprising logic in the first guest domain to receive a text list of the contents of the peripheral device.

5. The device of claim 1, further comprising logic to determine if an auto run file is on the peripheral device wherein the peripheral related task prevents the first guest domain from accessing the auto run file.

6. The device of claim 1, further comprising a blacklist of the peripheral device to prevent the first guest domain from accessing the peripheral device.

7. The device of claim 1, further comprising a scanner to scan the contents of the peripherai device for malicious code prior to allowing access to the peripheral device by the first guest domain.

8. The device of claim 1, further comprising an interface to select the level of access the first guest domain has to communicate with the peripheral device.

9. A method of communicating with a peripheral connected to a computing device comprising:

isolating a peripheral related task from a first guest domain;

generating a virtual appliance to execute the peripheral related task;

receiving, by the virtual appliance from the peripheral, the type of peripheral;

validating, by the virtual appliance, the type of peripheral; and

preventing communication with the peripheral by the first guest domain until the type of peripheral is validated.

10. The method of claim 1, scanning of the contents of the peripheral device for malicious code prior to allowing access to the peripheral device by the fsrst guest domain.

11. The method of claim 1, selecting the level of access the first guest domain has to communicate with the peripheral device.

12. The method of claim 1, determining if an auto run file is on the peripheral device wherein the virtual appliance prevents the first guest domain from accessing the auto run file.

13. A non-transitory computer readable medium comprising a privileged domain that if executed by a processor causes a computing device to:

isolate a peripheral related task from a first guest domain;

generate a virtual appliance to execute the peripheral related task;

receive, by the virtual appliance from the peripheral, the type of peripheral;

validate, by the virtual appliance, the type of peripheral; and

prevent communication with the peripheral by the first guest domain until the type of peripheral is validated.

14. The computer readable medium of claim 13 further comprising a peripheral related task that if executed causes a computing device to:

scan the contents of the peripheral device for malicious code prior to allowing access to the peripheral device by the first guest domain.

15. The computer readable medium of claim 13 further comprising a peripheral related task that if executed causes a computing device to:

select, through an interface, the level of access the first guest domain has to communicate with the peripheral device.

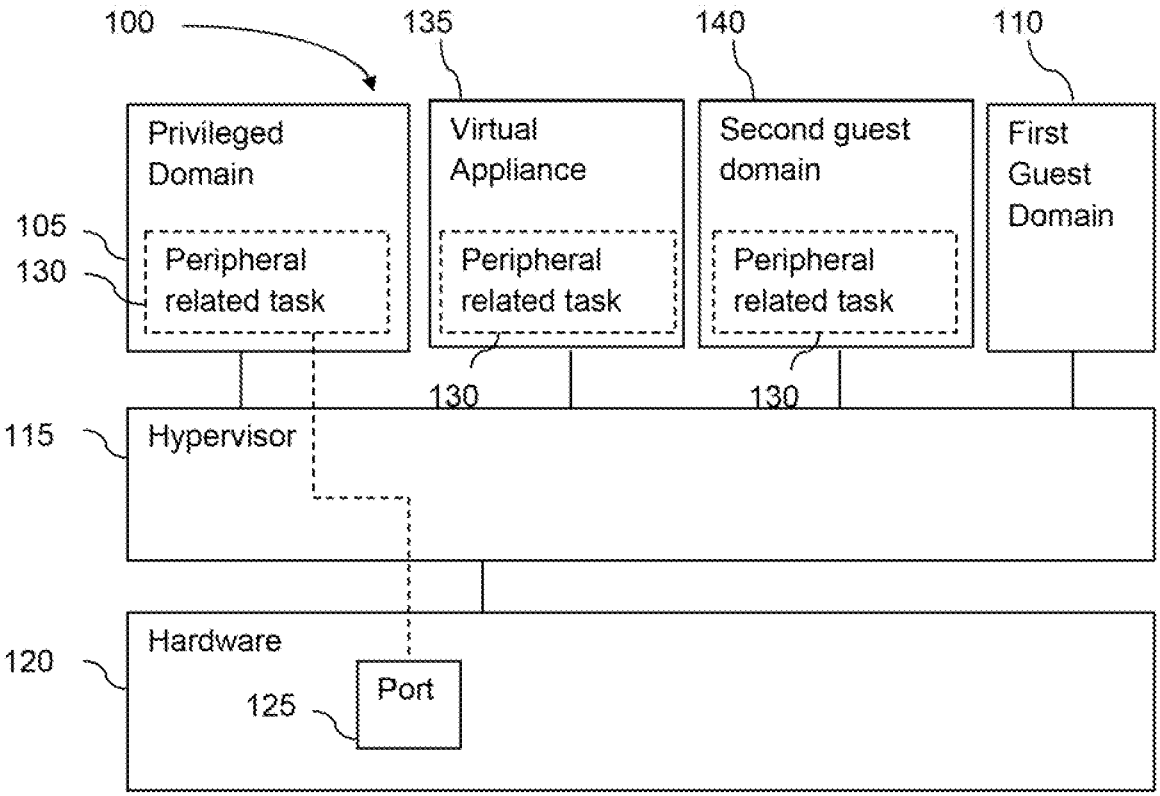
100
135
140
110
Privileged Domain
Virtual Appliance
Second guest domain
First Guest Domain
105
130
Peripheral related task
Peripheral related task
Peripheral related task
130
130
115
Hypervisor
120
Hardware
125
Port

FIG. 1

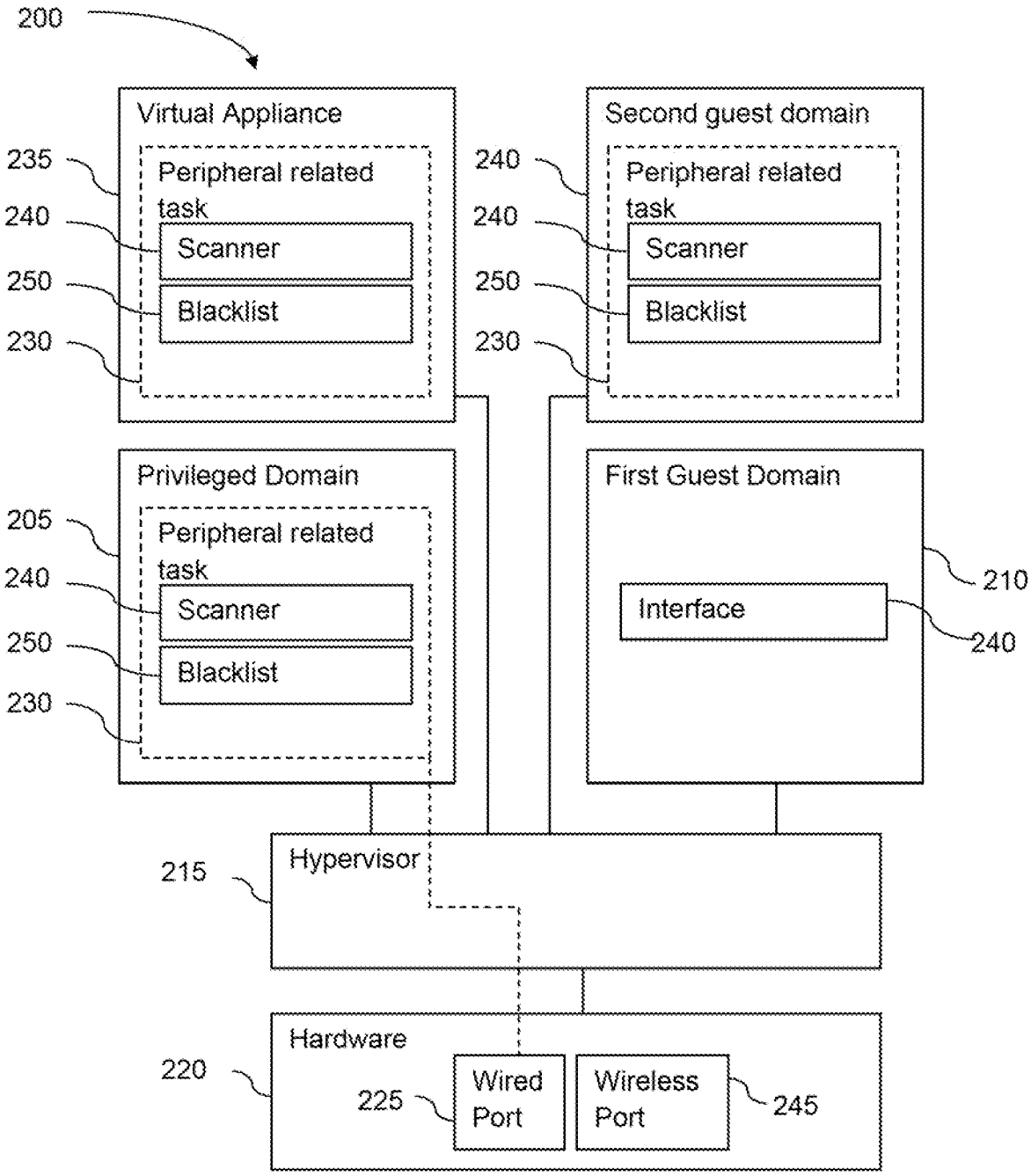
200
Virtual Appliance
235
Peripheral related task
240
Scanner
250
Blacklist
230
Second guest domain
240
Peripheral related task
240
Scanner
250
Blacklist
230
Privileged Domain
205
Peripheral related task
240
Scanner
250
Blacklist
230
First Guest Domain
210
Interface
240
215
Hypervisor
220
Hardware
225
Wired Port
Wireless Port
245

FIG. 2

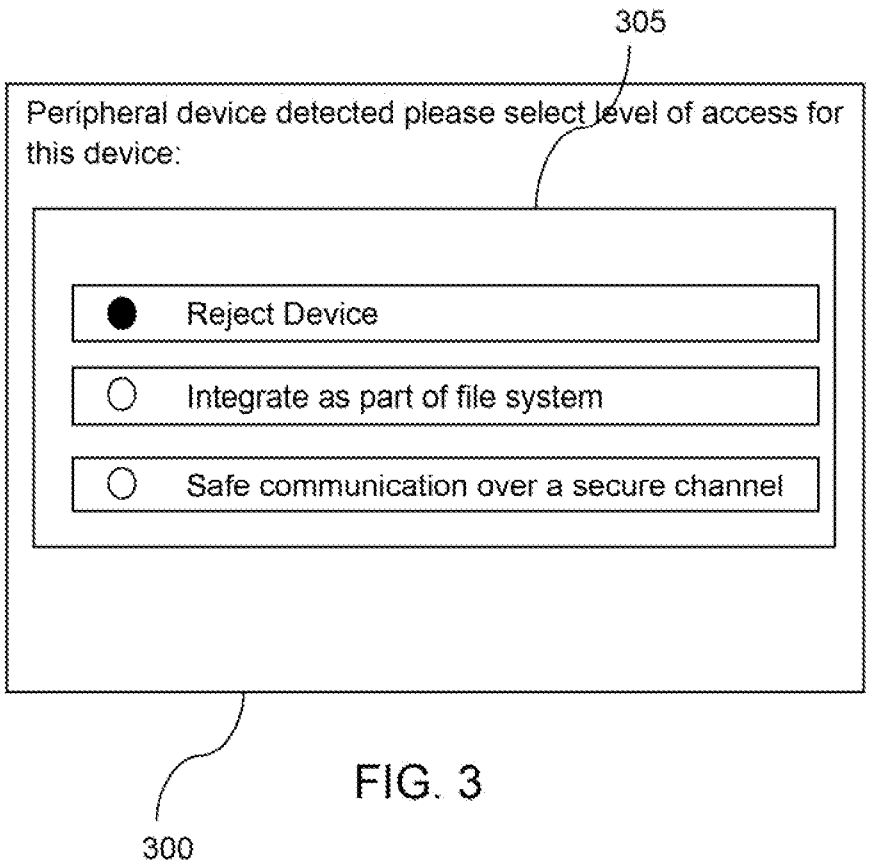
305
Peripheral device detected please select level of access for this device:
Reject Device
Integrate as part of file system
Safe communication over a secure channel
300

FIG. 3

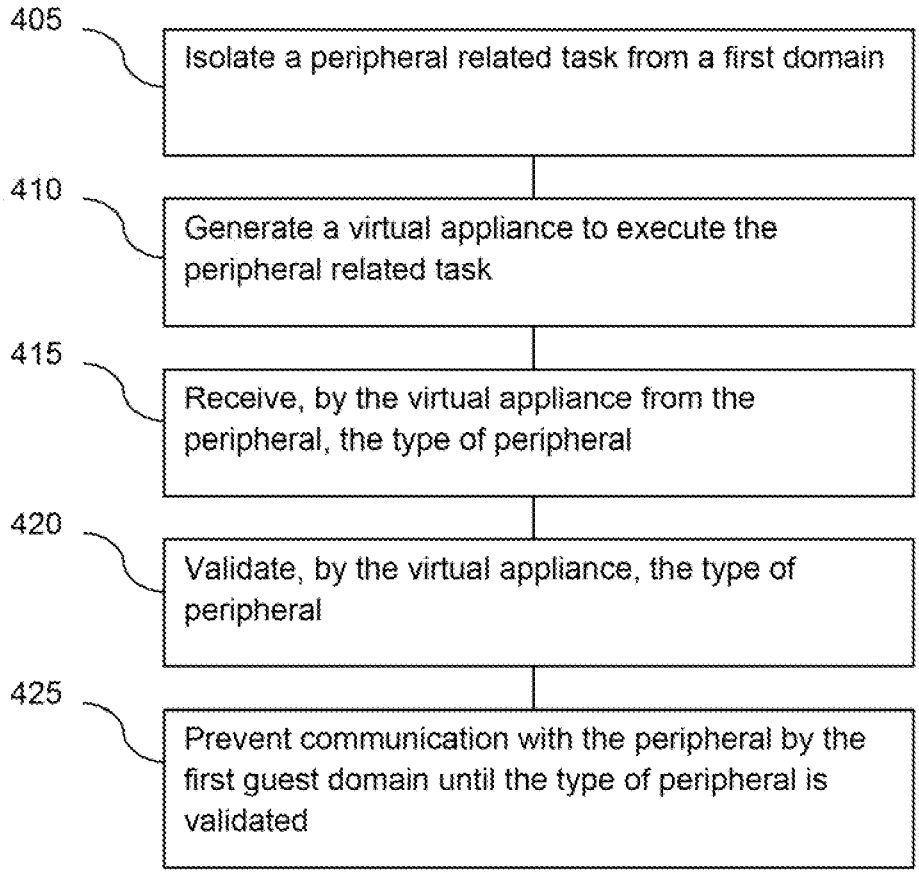
405
Isolate a peripheral related task from a first domain
410
Generate a virtual appliance to execute the peripheral related task
415
Receive, by the virtual appliance from the peripheral, the type of peripheral
420
Validate, by the virtual appliance, the type of peripheral
425
Prevent communication with the peripheral by the first guest domain until the type of peripheral is validated

FIG. 4

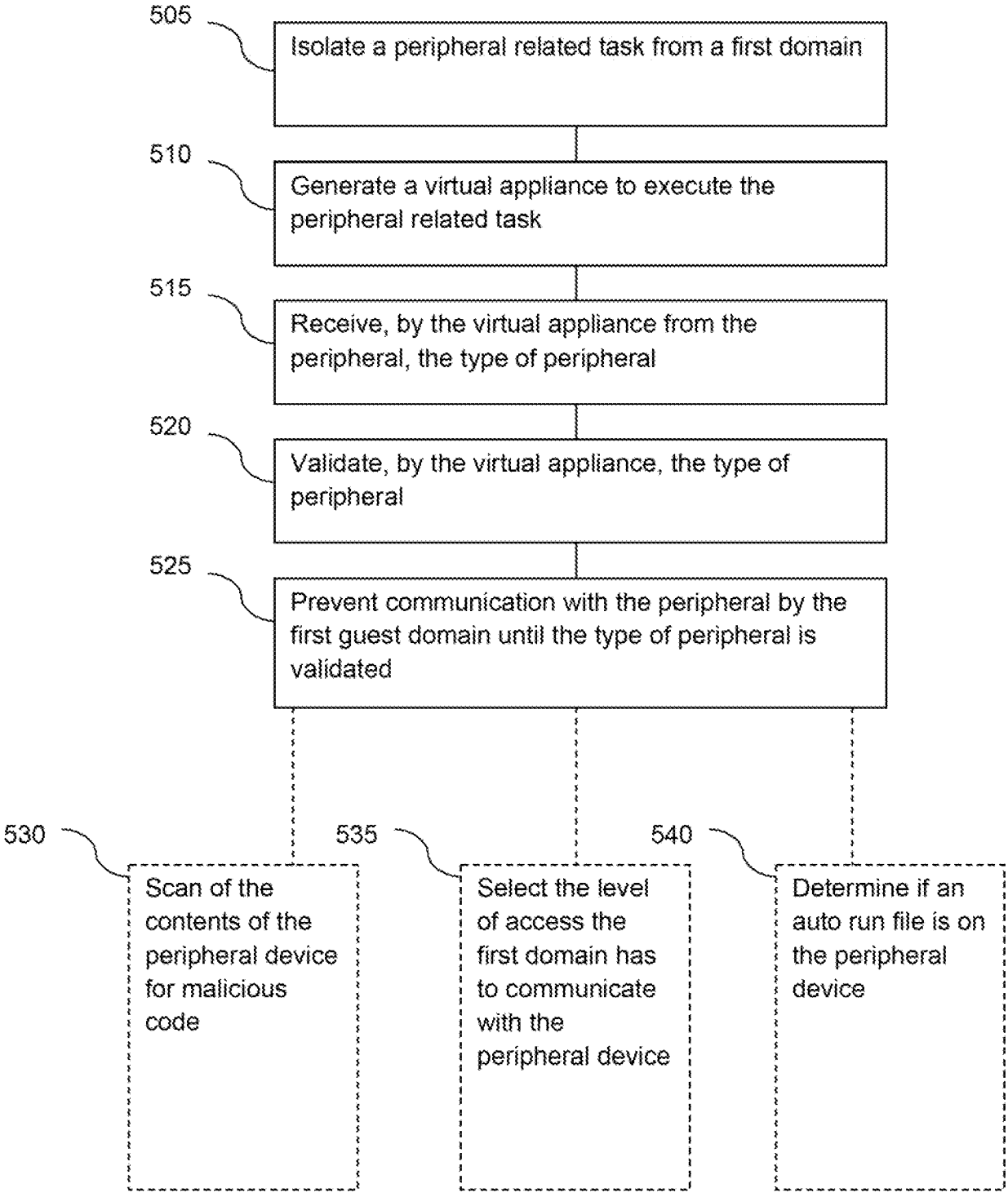
505
Isolate a peripheral related task from a first domain
510
Generate a virtual appliance to execute the peripheral related task
515
Receive, by the virtual appliance from the peripheral, the type of peripheral
520
Validate, by the virtual appliance, the type of peripheral
525
Prevent communication with the peripheral by the first guest domain until the type of peripheral is validated
530
Scan of the contents of the peripheral device for malicious code
535
Select the level of access the first domain has to communicate with the peripheral device
540
Determine if an auto run file is on the peripheral device

FIG. 5

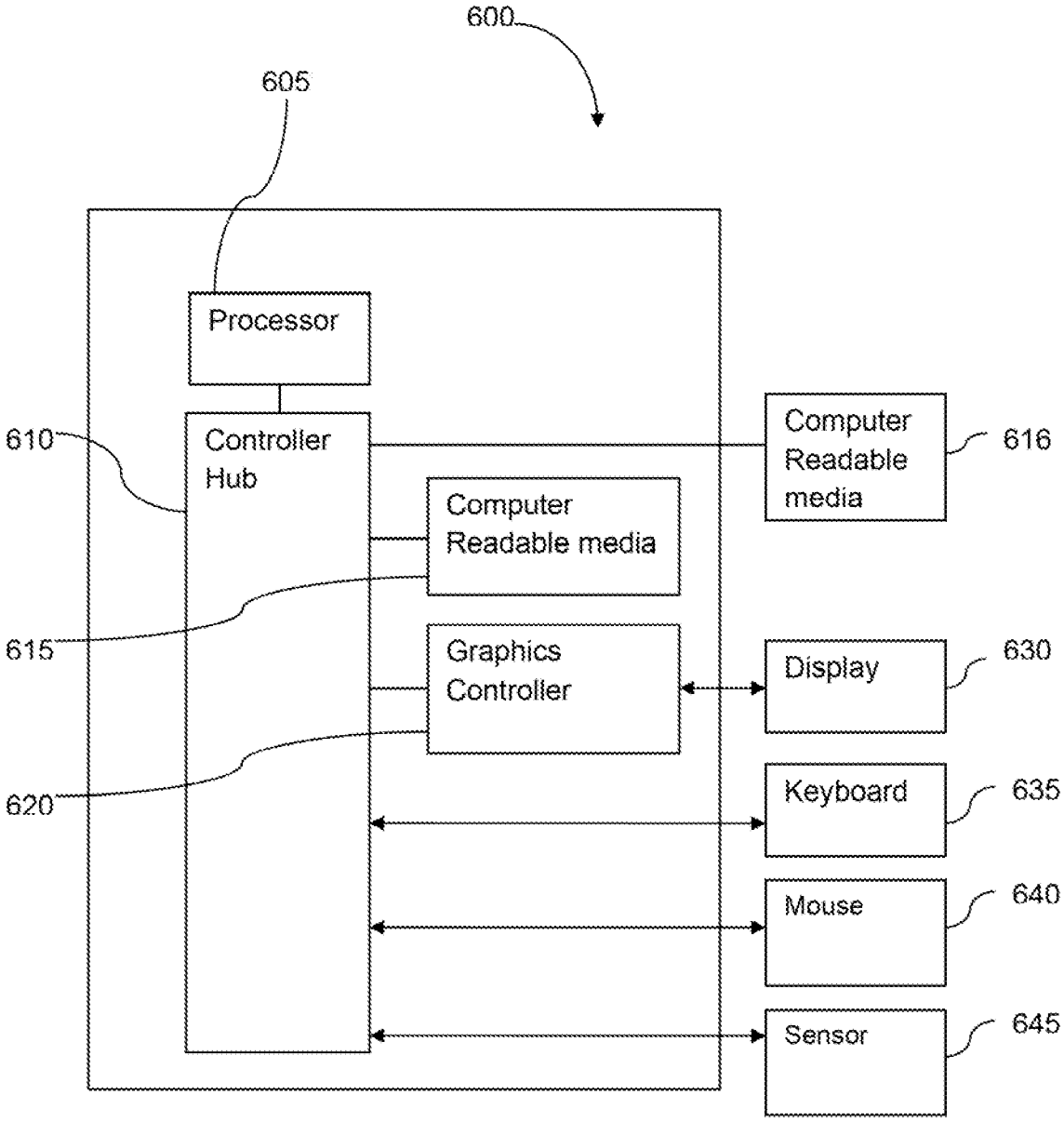
600
605
Processor
610
Controller Hub
615
620
Computer Readable media
616
Computer Readable media
Graphics Controller
Display
630
Keyboard
635
Mouse
640
Sensor
645

FIG. 6

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US20 11/043716

A. CLASSIFICATION OF SUBJECT MATTER

G06F 21/22(2006.01)i, G06F 9/06(2006.01)i

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 21/22; G06F 3/048; G06F 15/00; G06F 9/44; G06F 9/24; G06F 9/06; H04L 9/00; G06F 11/00

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

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

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

eKOMPASS(KIPO internal) & Keywords: virtual, machine, domain, peripheral, device, access, attack, malicious, code, malware, virus, guest, privilege, scan, isolate, prevent, hypervisor

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	KR 10-2009-0068833 A (SAMSUNG ELECTRONICS CO., LTD.) 29 June 2009 See abstract ; paragraphs [21] - [55] ; figures 1-4.	1-15
A	US 2010-0175108 Al (PR0TAS ANDRE) 08 July 2010 See abstract ; paragraphs [30] - [40] ; figure 1.	1-15
A	KR 10-2009-0100614 A (SAMSUNG ELECTRONICS CO., LTD.) 24 September 2009 See abstract ; paragraphs [60] - [90] ; figures 5-7.	1-15
A	KR 10-2011-0055391 A (INTERNATIONAL BUSINESS MACHINES CORP.) 25 May 2011 See abstract ; paragraphs [18] - [36] ; figures 1-2.	1-15

☐ 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 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	Date of mailing of the international search report
01 FEBRUARY 2012 (01.02.2012)	**09 FEBRUARY 2012 (09.02.2012)**
Name and mailing address of the ISA/KR Korean Intellectual Property Office Government Complex-Daej eon, 189 Cheongsa-ro, Seo-gu, Daejeon 302-701, Republic of Korea Facsimile No. 82-42-472-7140	Authorized officer Shin Sang Gil Telephone No. 82-42-481-8480

Form PCT/ISA/210 (second sheet) (My 2009)

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
KR 10-2009-0068833 A	29.06.2009	US 2009-0165133 A1	25.06.2009
US 2010-0175108 A1	08.07.2010	US 2010-019935 1 A1	05.08.2010
KR 10-2009-0 1006 14 A	24.09.2009	US 2009-0241 110 A1	24.09.2009
KR 10-201 1-0055391 A	25.05.2011	US 201 1-01 19669 A1	19.05.201 1