APPLICATION LEVEL ACCESS PRIVILEGE TO A STORAGE AREA ON A COMPUTER DEVICE

Field of the Invention

[0001] The present invention relates to processing of applications for use in a computer device, and more particularly, to the storing of applications in a file structure and limiting the access of applications to the storage area on a computer device.

Background

[0002] Wireless communication has experienced explosive growth in recent years. As consumers and businesses rely more on their wireless devices, such as mobile phones and personal digital assistants (PDAs), wireless service providers, i.e., carriers, strive to provide additional functionality on these wireless devices. This additional functionality will not only increase the demand for wireless devices but also increase the usage among current users.

[0003] The environment of a wireless device creates unique challenges when one considers the execution of application on a wireless device. Methods of downloading the applications and removing the applications need to be developed. In addition, there is a need for security on the wireless device. Security concerns on the wireless device include controlling the environment the best way possible so that an application cannot, intentionally or unintentionally, degrade or corrupt other files on the wireless device or the network on which the wireless device communicates.

[0004] Applications, during execution, may perform various operations such as reading, writing, and modifying files stored on the device, or accessible to the device. These files can be system files used by the device, such as device drivers, other application files or data files.

[0005] It is desirable control the application's access to other files accessible to the device as a security measure to limit any damage an application may have on the device, or other systems connected to the device. This damage may be intended via a virus on the application or may be unintended where the application executes with poorly written code that unintentionally dominates a resource, damages other applications or data.

[0006] Currently, the only method of controlling an applications access to storage area includes defining privilege levels to users. A user may be given a privilege level that allows the user to read, write or modify files in various storage areas. This mechanism only controls access at the user level.

[0007] This practice, however, does not allow the limiting of storage access based on the application executing. While the user itself may attempt to limit the application's access to storage, applications executed by the user are all given the same access to storage as is defined to the user. Furthermore, there is no mechanism to limit the application's access to storage outside of the control of the user.

[0008] Consequently, what is needed in the art is a system and method for limiting an application's access to storage. In addition, it is preferable that such limitation be established so that the device or user of the device be unable to change the access limitation imposed on the application.

SUMMARY OF THE INVENTION

[0009] Systems and methods consistent with the present invention overcome the shortcomings of existing systems by creating a safer environment for application storage and limiting access of the application to other storage areas. Furthermore, systems and methods consistent with the present invention provide a mechanism such that the device and/or user of the device be unable to change the access limitation imposed on the application.

[0010] In one embodiment, the present invention provides a method of executing an application on a device comprising the steps of receiving a request to execute the application, determining if the application was modified, in the event the application was not modified, mapping the application to a unique storage location, granting the application access to the unique storage location, and denying a second application access to the unique storage location.

[0011] In another embodiment, the present invention provides a device having a file structure, comprising a control program to store files within the file structure, wherein the control program is operable to store multiple applications within the file structure, each application having an associated portion of the file structure and the control program is operable to grant each application access to its associated portion and deny access to each application to other associated portions of the storage area.

[0012] In yet another embodiment, the present invention provides a method for storing an application on a device, comprising the steps of receiving the application at the device, storing the application in a storage on the device, and limiting access of the application to a unique portion on the storage, independent of a user access privilege on the device. Further embodiments of the present invention are also described in the following description and attached drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] The accompanying drawings, which are incorporated in and constitute a part of the specification, illustrate presently preferred embodiments of the invention and, together with the general description given above and the detailed description of the preferred embodiments given below, serve to explain the principles of the invention. In the drawings:

[0014] Fig. 1 is a block diagram depicting a computer device having storage in which an exemplary embodiment of the present invention may be practice;

[0015] Fig. 2 is a block diagram depicting a wireless system architecture containing wireless devices having storage and application access process in an exemplary embodiment of the present invention;

[0016] Fig. 3 is a block diagram depicting a file structure used in the implementation of an exemplary embodiment of the present invention;

[0017] Fig. 4 is a flowchart depicting the process of storing an application that limits access of an application to the storage area of a device in an exemplary embodiment of the present invention; and

[0018] Fig. 5 is a flowchart depicting the process of limiting access of application to a storage area in an exemplary embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[0019] Reference will now be made in detail to the presently exemplary and preferred embodiments of the invention as illustrated in the accompanying drawings, in which like reference characters designate like or corresponding parts throughout the several drawings. The nature, objectives and advantages of the present invention will become more apparent to those skilled in the art after considering the following detailed description in connection with the accompanying drawings.

Introduction

[0020] The present invention limits an application's access to a storage area on a computer device. The same user on a device may install multiple applications. The user may have access to all the applications and the storage area of each application. Systems and methods consistent

with the present invention, however, limit each of the application's access to areas of storage other than that allocated to the specific application. By defining this access per application and not per user, safer execution of applications can be achieved by reducing the unwanted consequence that may occur by an application overwriting or inappropriately accessing areas of storage that the application should not. This increased level of safety is provided even when the user, executing the application, has the rights to access the other storage areas that the application is prohibited from accessing.

[0021] Furthermore, systems and methods consistent with the present invention also provide for the access to shared areas of storage so that multiple applications may maintain their private and secure areas, but may also share files, such as data and libraries, with other applications.

[0022] It will be recognized to those skilled in the art that the forgoing describes an application file type being distributed and executed for simplicity of description. An "application" may also include files having executable content, such as: object code, scripts, java file, a bookmark file (or PQA files), WML scripts, byte code, and perl scripts. In addition, an "application" referred to herein, may also include files that are not executable in nature, such as documents that may need to be opened or other data files that need to be accessed.

[0023] Figure 1 is a block diagram depicting a computer device having storage in which an exemplary embodiment of the present invention may be practice. In this exemplary embodiment, a computer device 105 contains a storage area 110, a CPU 115 and an Input/Output (I/O) interface 120 interconnected via a bus 130. It will be recognized by those skilled in the art that these are but a few of the components that may be contained in a computer device. Furthermore, many other paradigms of interconnecting various components of a computer device may be used.

[0024] The computer device 105 may interface with outside systems via a network 100 and also may receive applications or other data via of the I/O device 125 (such as via CD-ROM, smart card, or floppy disk) into the I/O interface 120. The storage area 110 of the computer device 105 is used to store data and applications received into the computer device 105. The storage area 110 may comprise a hard disk drive, flash memory or other methods of storage (not shown). Furthermore, the computer device may store other information in storage area 110, such as operating system files, resource files, configuration files, and libraries. A separate storage area, such as ROM, EPROM and RAM, however, may also be used to store these or other files.

[0025] The computer device may be a personal computer with a fixed location when connected to the network 100 or may be a mobile wireless device that may change geographic location as it is communicating with the network 100.

[0026] The applications 135 may be received by the computer device via of the network 100 or via the I/O device 125. As described above, these applications include, but are not limited to, files having executable content, such as: object code, scripts, java file, a bookmark file (or PQA files), WML scripts, byte code, and perl scripts.

[0027] The network 100 may be a landline network incorporating private and public networks, such as the Internet or may incorporate an RF network, such as that used for wireless communication.

[0028] In one embodiment of the present invention, the computing device 105 may also contain a control program, such as the BREW API developed by QUALCOMM Incorporated. A control program can be used to coordinate the storage mechanism that provides the limiting access to the applications, as disclosed herein.

[0029] Figure 2 is a block diagram depicting a wireless system architecture containing wireless devices having storage and application access process in an exemplary embodiment of the present invention. A central server 202 is an entity that certifies, either by itself or in combination with a certification server, the application programs as compatible with a defined set of programming standards or conventions. As described earlier, these programming standards may be established so that the application will execute on a BREW™ software platform, developed by QUALCOMM Incorporated.

[0030] In one embodiment, the central server database 204 consists of a record of the identifications for each application program downloaded at any time onto each wireless device 230 in the network 200, an Electronic Service Number ("ESN") for the individual who downloaded the application program, and a Mobile Identification Number ("MIN") unique to the wireless device 230 carrying that application program. Alternatively, the central server database 204 contains records for each wireless device 230 in the network 200 of the wireless device model, wireless network carrier, the region where the wireless device 230 is used, and any other information useful to identify which wireless device 230 are carrying which application programs. In addition, the central server database may also store this developer identifying information associated with an application.

[0031] The central server 202 communicates with one or more computer servers 206, over a network 208, such as the Internet (preferably secured). The servers 206 also communicate with a

carrier network 210 via a network 208. The carrier network 210 communicates with the MSC 212 by both the Internet and POTS (plain ordinary telephone system) (collectively identified in Figure 2 as 211). The Internet connection 211 between the carrier network 210 and the MSC 212 transfers data, and the POTS 211 transfers voice information. The MSC 212, in turn, is connected to multiple base stations ("BTS") 214. The MSC 212 is connected to the BTS by both the Internet 211 (for data transfer) and POTS 211 (for voice information). The BTS 214 sends messages wirelessly to the wireless devices 230 by short messaging service ("SMS"), or any other over-the-air method.

[0032] The above network may be used to send an application to a computer device, such as the wireless device 230. The application, in one embodiment, has a unique identifier to distinguish it from other applications or files and a digital signature to detect modifications to the application. This digital signature may be bound to the application and stored on the wireless device either bound or separate, but still associated with, the application. The application is sent to the wireless device from the central server to one of the various servers 206 trough the MSC and BTS to the wireless devices 230.

[0033] Figure 3 is a block diagram depicting a file structure used in the implementation of an exemplary embodiment of the present invention. On a computer device having a storage area, one embodiment of the present invention creates a file structure to aid in limiting access of the applications to the storage areas. This structure in figure 3 depicts a hierarchical file structure, however, it will be recognized by those skilled in the art that many file structures may be used to implement systems and methods that limit an application's access to a storage area as describe herein.

[0034] In this embodiment, the root of the file structure defines the files to be stored in this portion of the storage area. Subdirectories to the File root directory 300 include Applications 305 and Shared 310. As the computer device receives applications, they are stored in subdirectories (e.g., App X 315 and App Y 320) under the Applications directory 305. It will be recognized by those skilled in the art that this hierarchical structure can be structured in many ways using various subdirectory schemes while still implementing the present invention.

[0035] In one embodiment of the present invention, the permissions associated with each of the applications are stored in the Applications directory 305. These permissions may be used to grant the application rights to specific files, such as APIs, data and libraries.

[0036] It is preferable that each application has a unique identifier associated with it and this unique identifier be used to create the individual applications' subdirectories under Applications

subdirectory 305. In one embodiment, each application has a unique name and the subdirectory is created using the name of the applications.

[0037] As each application is stored under the Application subdirectory 305, a mapping is created and associated with each application. This mapping prevents the application from accessing other areas of the file structure. In one embodiment, App X subdirectory 315 is mapped to a root directory with respect to App X's calls to the storage area. For example, from App X's perspective, it is stored at a root directory in the file structure even though in actuality, it is a subdirectory. In this embodiment, by mapping the App X subdirectory 315 to a root directory, calls from App X cannot access any of the files outside of the App X 315 directory. App X can, however, access the App X subdirectory 315 and any subdirectories (not shown) under the App X subdirectory 315. Consequently, under this implementation, App X cannot access the Files root directory 300, the Applications subdirectory 305 or, importantly, App Y subdirectory 320.

[0038] Similarly, within the same device accessed by the same user, when the computer device receives App Y, it is stored under the Applications subdirectory 305 in the App Y subdirectory 320. As with App X, a mapping is created with App Y such that it appears to App Y that it is stored at the root directory. This limits access of App Y to just the App Y subdirectory 320 and any subdirectories under it (not shown). This file structure and mapping scheme provides the advantage, similar as with App X, of preventing App Y from accessing other parts of the file structure.

[0039] Consequently on the computer device, applications may be granted access to files based on the specific applications and not based solely on the access privilege level of the user using the computer device.

[0040] Mapping the applications directory as a root directory in the file structure is one embodiment of the present invention and it will be recognize by those skilled in the art that other paradigms exist, such as semaphores, permission lists, and other techniques used to limit an applications access to the storage area.

[0041] In another embodiment of the present invention, it is anticipated that it may be desirable to share files, such as data or libraries, among applications. Therefore, in such instances, it is preferred to establish a shared directory 310 and include a mapping of the shared directory to each of the applications that want to share data. It is also preferable that this shared data be mapped to the applications such that it exists at a root directory, for example, of a separate drive. Mapping the shared directory 310 as a root directory will prevent each of the applications' access

to the parent directories of the shared directory and consequently to any subdirectories to the parent directories of the shared directory.

[0042] In addition, it is preferred that digital signatures are associated with the unique identifier, such as the application name described above. By using digital signatures, or some other modification detection technique, it may be detected whether an application modifies the unique identifier by which the subdirectory is created. The modification detection and/or prevention creates a safer system by providing an indication when an application is acting inappropriately, such as renaming itself the name of another application to access it's subdirectory. For example should App X rename itself App Y, and should the system allow a remapping based on the name, i.e., the unique identifier in this case, App X would then have access to App Y's files.

[0043] Figure 4 is a flowchart depicting the process of storing an application that limits access of an application to the storage area of a device in an exemplary embodiment of the present invention. The process begins by receiving an application having a unique identifier. (Step 400). The process may be controlled by a control program that provides some direction on the storage and mapping of the received applications. In one embodiment, the control program is a BREW™ API developed by QUALCOMM Incorporated running on a wireless device.

[0044] In one embodiment, the unique identifier is the application name. In this embodiment, applications sent to or received by the computer device need to be monitored such that the same application name is not sent to or received by the computer device. Other identifiers may be used and it is recognized by one skilled in the art how to monitor incoming applications to determine if a duplicate identifier exists.

[0045] The process proceeds to store the received application in a subdirectory in the file structure (Step 405). This may include creating the subdirectory using the unique identifier received with the application. The process then creates a mapping of the application subdirectory as root directory (Step 410). Consequently when the application is executing, it appears to the application that it is located at a root directory.

[0046] The process then proceeds to provide the application access to a shared directory by creating a mapping of another root directory to the shared directory (Step 415). This shared directory may be set up so that multiple applications have access to this directory for the sharing of files, such as data and libraries.

[0047] If the computer device receives a subsequent application (step 420), the process repeats but creates a separate subdirectory for the subsequent application based on the unique identifier. Consequently, it appears to the subsequent application that it is located at a root directory.

Furthermore, it may be mapped to the same shared directory as the previous application was mapped for the sharing of files.

[0048] If no additional applications are received, the process ends (Step 425).

[0049] Fig. 5 is a flowchart depicting the process of limiting access of application to a storage area in an exemplary embodiment of the present invention. The process begins by receiving a request to execute an application on a computer device (Step 500). This request may be made by a user via a user input or may be made by another application either on the computer device or connected to the computer device, such as via a network.

[0050] In one embodiment, the application has a digital signature associated with the application to detect modifications. In this embodiment, the process proceeds to evaluate the digital signature associated with the application (Step 505). If the application was modified, as determined by the digital signature the computer device may choose many courses of action, including any one or many of the following: denying execution of the application, informing the user of the modified application, notifying the computer device monitoring system of the modified application, or notifying a server or other computer system of the modified application.

[0051] If the application is not modified, the process creates access privileges to the application (Step 508). These privileges may be created by using the mapping process described above with respect to figures 3 and 4. Other privilege schemes, however, may be used to define privilege rights of the application to the storage areas on the device.

[0052] During the execution of the application, the application may receive a request from to access to a storage area (Step 510). For example, the application may request to perform a read, write or modify operation on a file located in the storage area. Such files may include those files that control resources of the computer device. The computer device determines if the application is granted privilege to the location of the file (Step 515). If the mapping paradigm is used above, the computer system determines the privilege by virtue of the fact that the application cannot access files above the root directory to which the application is mapped.

[0053] If the application has privilege to the file's location as determined in Step 515, then the application is granted access to the file (Step 520). If the application is not permitted access to the file's location, access is denied (Step 525) and the process ends (Step 530).

CONCLUSION

[0054] The foregoing description of an implementation of the invention has been presented for purposes of illustration and description. It is not exhaustive and does not limit the invention to

the precise form disclosed. Modifications and variations are possible in light of the above teachings or may be acquired from practicing of the invention. For example, the described implementation includes software but one embodiment of the present invention may be implemented as a combination of hardware and software or in hardware alone. The invention may be implemented with both object-oriented and non-object-oriented programming systems. Additionally, although aspects of the present invention are described as being stored in memory, those skilled in the art will appreciate that these aspects can also be stored on other types of computer-readable media, such as secondary storage devices, like hard disks, floppy disks, or CD-ROM; a carrier wave from the Internet or other propagation medium; or other forms of RAM or ROM. The scope of the invention is defined by the claims and their equivalents.

WHAT IS CLAIMED IS:

CLAIMS

[c1] 1. A method for limiting access of a storage area on a device to an application, comprising the steps of:

storing an application in a first portion of the storage area;

assigning privileges of the first portion to the application;

granting the application the privilege to access the first portion; and

denying the application access to a second portion of the storage area.

[c2] 2. The method of claim 1 further comprising the step of granting a user of the device access to the second portion of the storage area.

[c3] 3. The method of claim 1, wherein the storage area is located on a wireless device.

[c4] 4. The method of claim 1, wherein the storing of application is performed by application programming interface on the wireless device.

[c5] 5. The method of claim 1 further comprising the steps of:

storing a second applications in the storage area; and

granting the first application and the second application access a third portion of the storage area.

[c6] 6. A method for storing a plurality of applications on a device, comprising the steps of:

storing the plurality of applications on a storage area on the device, wherein each one of the plurality of applications is associated with a separate portion of the storage area;

providing each one of the plurality of applications the privilege to access one associated portion of the storage area; and

denying one of the plurality of applications access to one of the associated portions of the storage area.

[c7] 7. The method of claim 6 further comprising the step of allowing a user of the device to access one application having access to one associated portion of the storage area and one application denied to one associated portion of the storage area.

[c8] 8. The method of claim 6 further comprising the step of granting access of the plurality of applications to a shared portion of the storage area.

[c9] 9. The method of claim 6 wherein the device is a wireless device.

[c10] 10. A device having a storage area, comprising:

a file structure associated with the storage area;

a control program that directs the storage of an application on the device; and

the storage area having a first application allowed access to a first portion of the storage area and denied access to a second portion of the storage area.

[c11] 11. The device of claim 10 further comprising a second application allowed access to the second portion of the storage area and denied access to the first portion of the storage area.

[c12] 12. The device of claim 11 wherein the first application and the second application are operable of accessing a third portion of the storage area.

[c13] 13. A method of executing an application on a device, comprising the steps of:

receiving a request to execute the application;

determining if the application was modified;

in the event the application was not modified, mapping the application to a unique storage location;

granting the application access to the unique storage location; and

denying a second application access to the unique storage location.

[c14] 14. The method of claim 13, wherein the second application has access to a second storage location and a user has access to the unique storage location and the second storage location.

[c15] 15. The method of claim 13 further comprising the step of granting the application and the second application access to a shared storage location.

[c16] 16. The method of claim 13, wherein the device is a wireless device.

[c17] 17. A device having a file structure, comprising:

a control program to store files within the file structure, wherein the control program is operable to store multiple applications within the file structure, each application having an associated portion of the file structure and the control program is operable to grant each application access to its associated portion and deny access to each application to other associated portions of the storage area.

[c18] 18 The device of claim 17, wherein the file structure is a hierarchical structure and the control program is operable to map each application to indicate that the application is stored at a root directory.

[c19] 19. The device of claim 17, wherein the control program maps a plurality of the multiple applications to a shared portion of the file structure.

[c20] 20. A method for storing an application on a device, comprising the steps of:

receiving the application at the device;

storing the application in a storage on the device; and

limiting access of the application to a unique portion on the storage, independent of a user access privilege on the device.

[c21] 21. The method of claim 20, wherein the device is a wireless device.

[c22] 22. The method of claim 20, wherein the step of limiting access to the application is performed by mapping the unique portion on the storage as the application's root directory.

[c23] 23. A system of limiting an application's access to a storage area, comprising:

means for receiving the application at the device;

means for storing the application in a storage on the device; and

means for limiting access of the application to a unique portion on the storage, independent of a user access privilege on the device.

[c24] 24 A computer-readable medium containing computer executable instructions for limiting an application's access to a storage area that when executing perform a method comprising the steps of:

receiving the application at the device;

storing the application in a storage on the device; and

limiting access of the application to a unique portion on the storage, independent of a user access privilege on the device.

[c25] 25. A method for limiting access of a storage area on a device to an application, comprising the steps of:

assigning privileges of the first portion of the storage area to an application;

granting the application the privilege to access the first portion; and

denying the application access to a second portion of the storage area.