

tional authorizations in the command table can be granted based on an initial AccessAuths verification.

Subsequently the operating system checks if the AAS provides an authorization for executing the next privileged command. For example, when `accessauths=auth2`, is the authorization required to execute the privileged command `/usr/sbin/cmd2`, then the operating system checks the command table and determines that the authorization `auth2` and privileges represented by `innateprivs` are granted to the `cmd2` and the operating system executes `cmd2`. Since the command table is checked before running `cmd2`, and no authorizations are granted to the user, execution of the privilege command takes place one time and the privileged command is restricted for execution only from within `cmdA` and only during the execution of the privileged command. Consequently, these privileges cannot be used to run any other privileged command even from within the executable.

FIG. 2 is a flowchart for Authorized Authorization Set System 200. At step (202) a command is run. The command includes one or more privileged sub-commands and one or more ordinary sub-commands. An AAS corresponding to the command is in the command table and defines an authorization corresponding to each privileged sub-command included in the command. At step (204) the operating system determines if the user's role is assigned with the `accessauth` corresponding to the command. If the user's role is assigned with the `accessauth` corresponding to the command, then the operating system assigns the corresponding AAS to the command. The command then starts executing a sub-command (206, and the operating system determines if the sub-command is a privileged command (208). When the sub-command is a privileged sub-command, the operating system checks the command table (209) and determines if `accessauth` of the sub-command is part of the command's AAS in the command table (210). If so, the operating system runs the sub-command (212) If not, the operating system determines whether to execute the sub-command without privileges (211). If so, the sub-command is run without privileges. If not, AASS 200 goes to step 214. At step 214, the operating system determines if there are any additional sub-commands left in the command for execution. In case there are additional sub-commands, the operating system reads the next sub-command (206).

When the operating system detects an ordinary command (208) the operating system executes the ordinary command (214). When the operating system determines that there are no additional sub-commands to be executed, the command stops (212). The system administrator may modify the authorizations defined by AAS in command table.

The Authorized Authorization Set System may take the form of an entirely hardware embodiment, an entirely software embodiment or an embodiment containing both hardware and software elements. In accordance with an embodiment of the present invention, the invention is implemented in software, which includes, but is not limited to firmware, resident software, microcode, etc.

Furthermore, the Authorized Authorization Set System may take the form of a computer program product accessible from a computer-usable or computer-readable medium providing program code for use by or in connection with a computer or any instruction execution system. For the purposes of this description, a computer-usable or computer readable medium may be any apparatus that may contain, store, communicate, propagate, or transport the program for use by or in connection with the instruction execution system, apparatus or device.

The afore-mentioned medium may be an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor

system (or apparatus or device) or a propagation medium. Examples of a computer-readable medium include a semiconductor or solid-state memory, magnetic tape, a removable computer diskette, a random access memory (RAM), a read-only memory (ROM), a rigid magnetic disk and an optical disk. Current examples of optical disks include compact disk-read only memory (CDROM), compact disk-read/write (CD-R/W), DVD and blu-ray disk.

In the aforesaid description, specific embodiments of the present invention have been described by way of examples with reference to the accompanying figures and drawings. One of ordinary skill in the art will appreciate that various modifications and changes can be made to the embodiments without departing from the scope of the present invention as set forth in the claims below. Accordingly, the specification and figures are to be regarded in an illustrative rather than a restrictive sense, and all such modifications are intended to be included within the scope of present invention.

What is claimed is:

1. A computer implemented process for eliminating inherited privileges in a role based access control system control comprising:

modifying an operating system to access a command a table and to only run a sub-command when an appropriate authorization is in the command table;

modifying the role based access control system to eliminate inherited privileges;

entering a plurality of authorized access sets into the command table, the plurality of authorized authorization sets comprising a set of authorizations, each authorization corresponding to a privileged sub-command within a command;

responsive to receiving a request from a user to run a command, determining if the user is assigned an `accessauth` for the command;

responsive to determining that the user is assigned an `accessauth` for the command, determining if a sub-command is a privileged sub-command, and responsive to determining that the sub-command is a privileged sub-command, accessing the command table;

responsive to accessing the command table, determining if the `accessauth` of the sub-command is included in an authorized authorization set of the command; and

when the `accessauth` of the sub-command is included in the authorized authorization set of the command, running the privileged sub-command;

whereby a privileged sub-command is run only when the command is run and cannot be run by the user in any other context or at any other time.

2. A programmable apparatus for providing access control comprising:

a programmable hardware connected to a memory;

a program stored in the memory;

wherein the program directs the programmable hardware to perform the following series of steps:

modifying an operating system to access a command a table and to only run a sub-command when an appropriate authorization is in the command table;

modifying the role based access control system to eliminate inherited privileges;

entering a plurality of authorized access sets into the command table, the plurality of authorized authorization sets comprising a set of authorizations, each authorization corresponding to a privileged sub-command within a command;