

**AUTHORIZED AUTHORIZATION SET IN RBAC MODEL**

This invention was made with Government support under PERCS PHASE III, HR0011-07-9-0002. THE GOVERNMENT HAS CERTAIN RIGHTS IN THIS INVENTION.

**FIELD OF THE INVENTION**

The invention relates generally to security system for a computer system, and specifically to using a command table, a modified operating system, and a modified Role Base Access Control (RBAC) system to only allow sub-commands to be run in accordance with authorizations in the command table.

**BACKGROUND OF THE INVENTION**

A system administrator controls a user's access to the resources of a computer system by assigning access rights to the user in a security system. One such system is a Role-Based Access Control (RBAC) system. The RBAC uses authorizations, roles, and privileges to grant rights according to different levels of functionality for different classes of users. Roles are a set of functions unique to a particular class of users of the computer system, and multiple authorizations may be assigned to a role in order to allow users under that role to perform the requisite functions unique to the particular class of users. Privileges are a part of the RBAC system that provide fine granular control of the system functions. A user acquires privileges based on authorizations granted to their role. Regular users are allowed access to various functions when they have relevant privileges. Privileges are typically mapped to bit masks and are used in the kernel space of the operating system to achieve privileged function specific security controls.

A problem arises in the RBAC system in regard to assignment of privileges. In an RBAC system, a user runs a command that has various sub-commands in which some of the sub-commands are ordinary commands while others are privileged sub-commands. For a user to run the command, the user's role must have an authorization. When the user is authorized to run the command, the operating system will assign the command with all the privileges required for running each privileged sub-command within the command. For example, one possible RBAC system of authorizations and privileges is shown below:

---

cmdA:	accessauths = AuthABC innateprivs = privread, privexecute inheritprivs = priv1, priv2
cmd1:	accessauths = auth1 innateprivs = priv1
cmd2:	accessauths = auth2 innateprivs = priv2

---

As used herein command shall have the same meaning as process, program, shell script, or parent, and sub-command shall have the same meaning as sub-process, sub-program, script, or child.

Referring to the above example, cmdA requires an access authorization, AuthABC, to be assigned to the user in order for the operating system to run the command. Additionally, cmdA also requires that the privileges, innateprivs and inher-

itprivs, be assigned to the user so that the sub-commands can be run. Innate privileges are privileges assigned to the command when the operating system determines that the command has the proper authorization. Inherit privileges are privileges that a command passes on to its sub-commands.

In general, various commands run through multiple sub-commands for sequential execution. The sub-commands may be either ordinary commands or privileged commands. Ordinary sub-commands do not require any authorization in order to execute, while privileged sub-commands require that the user be authorized to execute each of the privileged subcommands. In an RBAC system, the command gains all of the accumulated authorizations needed to run each of the sequentially executed privileged sub-commands. Thus, when an authorization is assigned to a role, and correspondingly to the users associated with that role, those users are free to use the authorization from any context. In other words, a user with an authorization to execute a privileged sub-command could use the sub-command from any command, or directly from the command line. An sub-command executed by an authorized user run with privileges throughout its lifetime creates a security risk. Therefore, a need exists for a way to eliminate this security risk by restricting the execution of privileged sub-commands only in the context of the execution of the sub-command and only during the time the command actually runs the sub-command.

**BRIEF SUMMARY OF THE INVENTION**

The Authorized Authorization Set System comprising a modified operating system, a command table containing authorized authorization sets, and a modified RBAC security system, eliminates the need for inherited privileges that must be passed to subcommands in order for the command to run. The modified operating system accesses a table containing authorized authorization sets which identify the privileges for all subcommands within a command. The operating system assigns authorized authorizations to the process. When the process starts executing, it will be able to execute the sub-commands as the corresponding authorizations required to execute them have been assigned in the authorized authorization set of the process. Since no inherited privileges are assigned, a user cannot gain access to a subcommand in any other context or at any other time. Therefore, the Authorized Authorization Set System only executes privileged sub-commands in the context of the execution of the sub-command and only during the time the command actually runs the sub-command.

**BRIEF DESCRIPTION OF THE DRAWINGS**

The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself, as well as a preferred mode of use, further objectives, and advantages thereof, will be understood best by reference to the following detailed description of an illustrative embodiment when read in conjunction with the accompanying drawings, wherein:

FIG. 1 is a block diagram of an apparatus for providing access control in accordance with an embodiment of the present invention; and

FIG. 2 is a flowchart depicting a process for providing access control in accordance with another embodiment of the present invention.

**DETAILED DESCRIPTION**

FIG. 1 is a block diagram of an apparatus for providing access control in accordance with an embodiment of the