

present invention. Computer system 100 includes processor 102, main memory 104, mass storage interface 106, and network interface 108, all inter-connected by system bus 110. Those skilled in the art will appreciate that this system encompasses all types of computer systems: personal computers, midrange computers, and mainframes, and that many additions, modifications, and deletions can be made to this computer system 100. For example, computer system 100 may include a display, a keyboard, a cache memory, and peripheral devices such as printers (not shown).

Processor 102 that can be constructed from one or more microprocessors and/or integrated circuits. Processor 102 executes program instructions stored in main memory 104. Main memory 104 stores programs and data that computer system 100 may access to perform commands and sub-commands as explained in conjunction with FIG. 2. Further, a programmable hardware executes these program instructions. The programmable hardware may include, without limitation hardware that executes software based program instructions such as processor 102. The programmable hardware may also include hardware where program instructions are embodied in the hardware itself such as Field Programmable Gate Array (FPGA), Application Specific Integrated Circuit (ASIC) or any combination thereof.

Main memory 104 includes one or more application programs 112, data 114, operating system 116, command table 118, and RBAC 120. When computer system 100 starts, processor 102 initially executes the program instructions that make up operating system 116. Operating system 116 is a sophisticated program that manages the resources of computer system 100 for example, processor 102, main memory 104, mass storage interface 106, network interface 108, and system bus 110.

Processor 102 under the control of operating system 116 executes application programs 112. Application programs 112 can be run with program data 114 as input. Application programs 112 can also output their results as program data 114 in main memory 104.

Mass storage interface 106 allows computer system 100 to retrieve and store data from auxiliary storage devices such as magnetic disks (hard disks, diskettes) and optical disks (CD-ROM). These mass storage devices are commonly known as Direct Access Storage Devices (DASD) 118, and act as a permanent store of information. One suitable type of DASD 118 is floppy disk drive that reads data from and writes data to floppy diskette 120. The information from DASD 118 can be in many forms. Common forms are application programs and program data. Data retrieved through mass storage interface 106 is usually placed in main memory 104 where processor 102 can process it.

While main memory 104 and DASD 118 are typically separate storage devices, computer system 100 uses well known virtual addressing mechanisms that allow the programs of computer system 100 to run smoothly as if having access to a large, single storage entity, instead of access to multiple, smaller storage entities (e.g., main memory 104 and DASD 118). Therefore, while certain elements are shown to reside in main memory 104, those skilled in the art will recognize that these are not necessarily all completely contained in main memory 104 at the same time. It should be noted that the term "memory" is used herein to generically refer to the entire virtual memory of computer system 100. In addition, an apparatus in accordance with the present invention includes any possible configuration of hardware and software that contains the elements of the invention, whether the apparatus is a single computer system or is comprised of multiple computer systems operating in concert.

FIG. 1 further depicts network interface 108 that allows computer system 100 to send and receive data to and from any network connected to computer system 100. This network may be a local area network (LAN), a wide area network (WAN), or more specifically Internet 122. Suitable methods of connecting to a network include known analog and/or digital techniques, as well as networking mechanisms that are developed in the future. Many different network protocols can be used to implement a network. These protocols are specialized computer programs that allow computers to communicate across a network. TCP/IP (Transmission Control Protocol/Internet Protocol), used to communicate across the Internet, is an example of a suitable network protocol.

FIG. 1 further depicts system bus 110 that allows data to be transferred among the various components of computer system 100. Although computer system 100 is shown to contain only a single main processor and a single system bus, those skilled in the art will appreciate that the present invention may be practiced using a computer system that has multiple processors and/or multiple buses. In addition, the interfaces that are used in the preferred embodiment of the present invention may include separate, fully programmed microprocessors that are used to off-load compute-intensive processing from processor 102, or may include I/O adapters to perform similar functions.

In a UNIX™ based operating system, a command can be written for any purpose. For example, the command, cmd A, has an authorized authorization set (AAS) associated with it. As used herein authorized authorization set (AAS) means a plurality of authorizations entered by a system administrator into a command table accessible by an operating system. The command can be represented as:

---

```

cmdA:
    accessauths = AuthABC
    innateprivs = privread, privexecute
    AAS = auth1, auth2, auth3

cmd1:
    accessauths = auth1
    innateprivs = priv1

cmd2:
    accessauths = auth2
    innateprivs = priv2

cmd3:
    accessauths = auth3
    innateprivs = priv3
  
```

---

As represented above, the AAS has three authorizations auth1, auth2, and auth3 corresponding to the three privileged commands: cmd1, cmd2, and cmd3 respectively. The AAS is entered into the command table by the system administrator. Access to the command is governed by the modified RBAC system that does not contain inherited privileges. The operating system determines that a user's role has the proper accessauth in order to run the command in accordance with the RBAC system. Innate privileges are assigned in accordance with the modified RBAC system. After that, the modified operating system controls access in accordance with the command table containing the AAS for the command and the authorizations for each sub-command.

In this example, in accordance with the AAS in the command table, cmdA will be assigned with all authorizations specified in the AAS. When cmdA executes cmd1, the AAS will be checked and if auth1 access is allowed to all privileges (priv1 from innateprivs) are assigned to cmd1. In this way the user's role need not have all the authorizations required to execute the privileged sub-commands in the command. Addi-