

Authentication Server Installation and Maintenance

David King

Information Technology Center

1. Installation

To get an Authentication Server running on a new cluster machine, two things are needed: the software binaries, and the data bases. There are four programs which make up the Authentication Server software:

- o AuthServer - The Authentication Server itself.
- o aamaint - The user/account maintenance utility.
- o aatest - The simple tester.
- o buildtbl - The basic database creation program.

The database consists of seven files:

- o adb - The Accounting Data Base (present only on the central machine).
- o broadcast.queue - The queue for messages to be sent to other clusters.
- o cluster.table - The list of all the cluster machines.
- o log - The log kept of server activity.
- o rates.table - The billing rate table.
- o tranfile - The billing transaction file.
- o uaf - The User Authorization File.

A single directory, perhaps /vice/auth or /vicebin/auth, must be created to contain all of the programs and all of the databases. It should be owned by a user responsible for it. The mode should be set so that only maintainers can read or write the files, if the system is not secure from users; mode 770 makes the most

July 18, 1984

sense.

To generate the Authentication Server programs, do the following in a temporary directory.

```
rcslink vice/auth
rco 'rcsls'
make
make aamaint
make buildtbl
```

This will compile the Authentication Server, the maintenance program, and the table-building program. Move AuthServer and aamaint to the database directory ('make install' will do that to /vice/auth). The mode should be set to 770 so that users can not read AuthServer.

To create the database files, connect to the directory and run the program buildtbl. buildtbl will then create the cluster table, rate table, and UAF with a standard set of parameters, and will create the broadcast queue, log, and transaction file as empty files. It will also create the ADB if its built-in table says that this machine is the central machine. At this point you should change the cluster table, etc., to whatever are the standard initial files in use.

To start the server running, simply connect to the database directory and run AuthServer with no arguments, while logged in as *root* or as a user of the same group as the owner. This will immediately return, after spawning off the actual server process in a background fork.

Once the server is running, you will need to fill in the User Authorization File and Accounting Data Base with the user entries you want, using the aamaint program; the best idea, if there is a cluster already set up the way you want it, is to use the 'j' command on the existing cluster to build a command file, and then the 'l' command on the empty cluster.

2. Maintenance

The program aamaint is used to add and delete users and accounts, and also to display transactions in the transaction file. It must be run on the cluster machine; its functions are not permitted to be sent in from workstations. To run it, log in to any user who has access to the aamaint program, connect to the database directory, and run aamaint.

The program prompts with the character "*>*". It accepts command lines consisting of a single-character command code, followed by any appropriate arguments, followed by newline; it has a very crude parsing system. Each command parses its arguments, makes an RPC connection to the local Authentication Server, sends a single request over the connection, closes the connection, and

July 18, 1984

reports if it was successful. The commands are:

- a "Add a user:" This will add a new user; the arguments are
- username The internal username of the user to be added.
 - password The initial password of the user, which can be up to 16 ASCII characters.
 - accounts The list of account numbers and quotas to be given. This is a series of pairs:
 - account The account number to be given.
 - quota The decimal usage allocation to be given for that account.
 - options Two options may be specified.
 - d account To specify a default account.
 - p time To specify a timestamp for the last password change. This is intended to be used only in join decks written by the 'j' command.
 - c To say that the password given was the 16 hex characters of the crunched password, and should not be considered an ASCII password to be crunched again. This is intended to be used only in join decks written by the 'j' command.
- d "Delete a user:" The argument is the internal username of the user to be deleted. The user's entries in the User Authorization File and Accounting Data base are deleted.
- n "Add account:" This will add accounts to an existing user; the arguments are
- username The internal username of the user to be given accounts.
 - accounts The list of account numbers and quotas to be given. This is a series of pairs:
 - account The account number to be given.

July 18, 1984

- quota The decimal usage allocation to be given for that account.
- r "Remove account:" This will remove accounts from a user. The arguments are
- username The internal username of the user to be given accounts.
- accounts The list of account numbers to be removed. This is a series of:
- account The account number to be removed.
- c "Change account quota:" This will change the account quotas of specified accounts. The arguments are
- username The internal username of the user to be changed.
- accounts The list of account numbers and quotas to be changed. This is a series of pairs:
- account The account number to be changed.
- quota The decimal usage allocation to be given to that account.
- q "Quit:" Use this command to exit from aamaint.
- s "Show:" Given a username, this command displays significant information from the user's entry in the User Authorization File.
- l "Load:" Given a file name, this command reads the file and processes each line as a command.
- j "Joindeck:" Given a file name, this command reads the User Authorization File and Accounting Data Base and writes a file of 'a' commands representing each of the users currently defined. This is useful for transferring an entire set of user definitions from one cluster to another, or for backup, or for providing to the administration's billing system's *fulljoins* process to reconcile differences.
- t "Transaction file:" This command interprets and displays all the transactions in the transaction file.

July 18, 1984