Imperium Security

Imperium is currently designed to be a "single processor" game. By that I mean that it does not support the ability to have the server talk to clients on machines on other systems via direct access to sockets or the like. However, it may be possible to do this if one machine is able to mount the file system on which the Imperium server creates it's FIFOs (such as via NFS).

In any case, the "prefered" way of making an Imperium game available via the net is to add a properly set up "ConImp" entry to your TCP/IP "services" file, such that ConImp gets started up with features like the "log" command disabled, and maybe to select a certain game (if you run more than one). I would expect you will need to be root to do this. Once you have it set up this way there is no problem with people using "telnet" or some similar program to start up an Imperium session.

To prevent other people from reading the Imperium data files directory you need to be sure of a few things:

1. Always make the Imperium server SUID to the owner of the Imperium programs. The server will take care to create all files with a umask of 077, so other people will not be able to read them.
2. Always start the Imperium server in a directory that is mode 700. This assures that even files like the log will not be in a place where others can see them.
3. Do not change the part in the Makefile's where a "fifo" and "fifo/servers" directory is created. By keeping the files seperated this way, and at the permissions that Imperium wants, you will prevent casual (normal user) access to the data files as well as "spoofing" of the Imperium server by user programs.
4. Read the instructions below on modifying the "ic.allow" and "is.allow" files.

Setting up the "ic.allow" and "is.allow" files

Previous versions of Imperium depended on the execution permissions of the ImpShut & ImpCtrl binaries to prevent unauthorized people from running reports or shutting down ImpServ.

The problem was that many times it was desireable to allow CERTAIN people to use CERTAIN commands, while preventing them from using other commands. The "ic.allow" and "is.allow" files follow a common format that allows you to specify default permissions, as well as per-user permissions/restrictions.

Here is an example file:

```
# This file determines which users may use which ImpCtrl commands.
#
impown all
all p
jjones none
```

Comments in this file can be created by starting the line with a "**#**". Basically the format consists of a user name in the first column, a single space, and then the list of command letters (the same ones you would use as command-line options) that that user is allowed to use.

You may use the username "all" to represent any user who is not listed specifically (any user who IS listed will have their commands limited to whatever appears after their username).

Specifying the command "all" (or "*") will allow that user to use ALL the command-line options. Specifying the command "none" (or "!") will prevent that user from using ImpShut/ImpCtrl at all.

So in the above example, "impown" is allowed to use all the commands, the rest of the users are limited to the "p" command, and "jjones" is not allowed to use any commands.

The "ic.allow" file controls access to ImpCtrl, and the "is.allow" file controls access to ImpShut. You do NOT have to have the same people listed in both of them. If either file is missing or unreadable, NOBODY will be able to run the corresponding program, including "root".