by Jim Fox

Root with Restrictions

***Jim Fox** works as a systems programmer for the University of Washington. He writes and maintains distributed applications that run on a variety of UNIX systems–and some non-UNIX ones. He is also the deputy manager for the Interoperability Project for SHARE's Open Systems Group. Email:* `fox@cac.washington.edu`.

wizard's apprentice

super user

wizard

Q: I've a question regarding the Q&AIX article that appeared in *SunExpert*, May 1997 ("setuid Dangers," Page 66), where Bud Ding Hacker finds a way to be root. Do you know of a way to set up a user with some root privileges such as executing commands and scripts so that a computer operator can run a system backup, for instance, or execute some commands that require one to be root? But, I don't want them to have complete root access, to overwrite or delete files, for example.

Raju Shah
TTI Inc.

A: A limited root capability is such a good idea one wonders why vendors haven't implemented it. Fortunately, someone else did, so you're saved a lot of work. But before we get to the program, let's consider the problem.

There's a reason some commands and files are restricted to root access. They have the capability to compromise the system. Allowing someone to run backup, for example, gives that person access to all files on the system, including `/etc/security/passwd` and the company president's email. Only give partial root access to people you already trust.

What you're really gaining is convenience. Restricted access prevents conscientious operators from accidentally causing damage, and it makes browsing of private files sufficiently difficult that it will not be done casually or whimsically. But don't be misled: Don't give this capability to anyone you don't know and trust.

The sudo Program

The program I have in mind is `sudo`. It was written by several generous people, most recently, Todd Miller of Courtesan Consulting, and is found at `http://www.courtesan.com/courtesan/products/sudo/`.

Get the distribution from one of the mirror sites shown on that Web page. You get source in a compressed `tar` file. Uncompress and untar it. Before building the program, there's some customization you will have to do. All of these are found in the include file `options.h` and are described more fully in the `OPTIONS` text file. Make the following changes to the defaults:

1. Define **`NO_MESSAGE`**. `sudo` has a kind of chummy, informal approach that is incompatible with its serious utility.
2. Define **`SECURE_PATH`**. By default, `sudo` allows a user to have any path at all.

That's intolerable. Check the path you assign to SECURE_PATH to make sure it contains only the necessary directories. An example of a good path might be /bin:/usr/sbin:/etc.

3. Define **NO_ROOT_SUDO**. This prevents root from running sudo. There's no reason for root to be running this program, and it invites chicanery to allow it.

4. Define **LOGGING** and **LOGFAC**. These tell sudo how to log usage. The defaults are OK, but you can't watch things without knowing what these values are.

Apart from these, the defaults won't hurt you. Even if this program is just for you, you really should avoid the funny error messages triggered by the USE_INSULTS definition. It may have been funny to hear HAL say, "I'm sorry Dave, I can't do that," in the movie *2001: A Space Odyssey*, but you won't like it when the hour is late, the raptors are chewing at the door, and your only hope of salvation keeps responding, "Whoa buddy, my horse types better'n that!" Computer humor has to be very innocent and subtle, or it doesn't work at all.

After setting and checking the options, run configure. This will build a Makefile. I always enjoy watching configure run. It gives a kind of bird's-eye view of the system.

sudo has some capability to use many common authentication protocols in addition to the standard password file. These include one-time passwords, SecurID cards, Kerberos and DCE. Activate one of these with the appropriate configure command-line option. See the INSTALL notes for details, if you want to use one of these methods.

Finally, type make install to install the files. The sudo program will be installed into /usr/local/bin with the setuid bit set.

Authorizing Users

Now you're ready to authorize some users. Do that by editing the authorization file /etc/sudoers. It contains a list of all users who have access to sudo and specifies what commands each user can run. A sample file is shown in Figure 1.

You have quite a bit of flexibility. The host specifications allow a single file to serve several machines for convenience. The sample permits the regular operators, joe and bob, to run all of the specified commands. The apprentice operator, pete, is allowed to do backups on all systems, but restores and shutdowns only on the server systems.

Obviously, I could not think of any really useful commands. The configuration is largely dependent on your local practices. You can get the idea though.

The sudoers file is actually a little more capable than I have shown–too capable, I think. Be conservative here. Look at the sudoers man page to see a more complete description of the authorization file.

sudo comes with an editor wrapper, visudo, which locks the file before running vi and verifies the contents afterward. This seems like overkill to me, but there's certainly no harm in using it.

Once sudo and the configuration file are in place, you're done. Any authorized user can run one of the allowed commands as root by typing

```
$ sudo command args
```

There are some optional parameters to sudo, but the plain way is probably how you'll be using it. You might want to keep an eye on the logs for your own peace of mind.

Well, now you can give limited permission to your operators to do backups. Be careful though. That limited permission can be hard to control.

Figure 1. Sample sudo Authorization File

```
# Hosts
Host_Alias  SERVERS=alpha,beta,gamma
Host_Alias  ADMIN=delta,chi

# Users
User_Alias  OPS=joe,bob
User_Alias  PRENTICE=pete

# Commands
Cmnd_Alias  LS=ls,cat
Cmnd_Alias  DOWN=shutdown,reboot,halt
Cmnd_Alias  BKP=backup.restore

# Who and what
OPS         LS,DOWN,BKP
PRENTICE    SERVERS=LS,BKP: ALL=backup

%wheels     DOWN
```