

NETWORK SECURITY HACKS™

100 Industrial-Strength Tips & Tools



O'REILLY®

Andrew Lockhart

HACK
#99

Scan for Root Kits

Use *chkrootkit* to determine the extent of a compromise.

If you suspect that you have a compromised system, it is a good idea to check for root kits that the intruder may have installed. In short, a root kit is a collection of programs that intruders often install after they have compromised the root account of a system. These programs will help the intruders clean up their tracks, as well as provide access back into the system. Because of this, root kits will sometimes leave processes running so that the intruder can come back easily and without the system administrator's knowledge. This means that some of the system's binaries (like *ps*, *ls*, and *netstat*) will need to be modified by the root kit in order to not give away the backdoor processes that the intruder has put in place. Unfortunately, there are so many different root kits that it would be far too time-consuming to learn the intricacies of each one and look for them manually. Scripts like *chkrootkit* (<http://www.chkrootkit.org>) will do the job for you automatically.

In addition to detecting over 50 different root kits, *chkrootkit* will also detect network interfaces that are in promiscuous mode, altered *lastlog* files, and altered *wtmp* files. These files contain times and dates of when users have logged on and off the system, so if they have been altered, this is evidence of an intruder. In addition, *chkrootkit* will perform tests in order to detect kernel module-based root kits. C programs that are called by the main *chkrootkit* script perform all of these tests.

It isn't a good idea to install *chkrootkit* on your system and simply run it periodically, since an attacker may simply find the installation and change it so that it doesn't detect his presence. A better idea may be to compile it and put it on removable or read-only media. To compile *chkrootkit*, download the source package and extract it. Then go into the directory that it created and type *make*.

Running *chkrootkit* is as simple as just typing `./chkrootkit` from the directory it was built in. When you do this, it will print each test that it performs and the result of the test:

```
# ./chkrootkit
ROOTDIR is '/'
Checking `amd'... not found
Checking `basename'... not infected
Checking `biff'... not found
Checking `chfn'... not infected
Checking `chsh'... not infected
Checking `cron'... not infected
Checking `date'... not infected
Checking `du'... not infected
```

```
Checking `dirname'... not infected
Checking `echo'... not infected
Checking `egrep'... not infected
Checking `env'... not infected
Checking `find'... not infected
Checking `fingerd'... not found
Checking `gpm'... not infected
Checking `grep'... not infected
Checking `hdparm'... not infected
Checking `su'... not infected
```

That's not very interesting, since the machine hasn't been infected (yet). *chrootkit* can also be run on disks mounted in another machine; just specify the mount point for the partition with the `-r` option, like this:

```
# ./chrootkit -r /mnt/hda2_image
```

Also, since *chrootkit* depends on several system binaries, you may want to verify them before running the script (using the [Tripwire \[Hack #97\]](#) or [RPM \[Hack #98\]](#) methods). These binaries are `awk`, `cut`, `egrep`, `find`, `head`, `id`, `ls`, `netstat`, `ps`, `strings`, `sed`, and `uname`. However, if you have known good backup copies of these, you can specify the path to them by using the `-p` option. For instance, if you copied them to a CD-ROM and then mounted it under `/mnt/cdrom`, you would use a command like this:

```
# ./chrootkit -p /mnt/cdrom
```

You can also add multiple paths by separating each one with a `..`. Instead of maintaining a separate copy of each of these binaries, you could simply keep a statically compiled copy of BusyBox handy (<http://www.busybox.net>). Intended for embedded systems, BusyBox can perform the functions of over 200 common binaries, and does so using a very tiny binary with symlinks. A floppy, CD, or USB keychain (with the read-only switch enabled) with *chkrootkit* and a static BusyBox installed can be a quick and handy tool for checking the integrity of your system.