

BORIS LOZA

finding trojans for fun and profit



Boris Loza, Ph.D., is a founder of Tego System Inc. and HackerProof Technology, in addition to being a contributor to many industry magazines. He holds several patents and is an expert in computer security.

■ bloza@hackerproofonline.com

“THE TROJAN HORSE” ORIGINALLY referred to the ploy used by the ancient Greeks to attack the city of Troy. Today, it’s fairly common knowledge that a trojan horse is an application a cheeky hacker tries to install on your hard disk to get easy access to your computer. A trojan can be part of a rootkit while masquerading as a legitimate application such as ls, df, or ps. In this article I will show you how to find rootkits and trojans using handy little utilities and a couple of tricks.

Checking Inodes

One of the ways to find trojan files in a current directory is to check inode numbers. Many rootkits modify the access and modification time of the files they replace, so at a glance a file may appear to be unchanged or even untouched. What remains is to check an inode number of a file in question.

Most installs will install files sequentially. For example, the output below shows inode numbers for files in the /etc directory:

```
$ ls -ai /etc | sort | more
.....
183491 TIMEZONE
183492 autopush
183493 cfgadm
183494 clri
183495 crash
183496 cron
.....
```

The `-i` option of `ls` lists the files’ inode numbers. As you can see from the output, most of the inode numbers are in sequence.

A broken number sequence indicates the possibility that those files were installed after the main installation took place. Look for out-of-place entries, either very high or very low. Also look for new groupings, as all the rootkit pieces were probably installed at the same time.

Note: The `newfs` command uses `fsirand(1M)` to install random inodes when creating a new file system. Also, if you use `fsirand` periodically, your system inode numbers will not be in sequence. For this reason, you may want to create a master database of all inode numbers for all your files. You can use something like the following to collect this information into a file:

```
# ls -aiR / > my_inodes
```

Put this database aside and check the inode numbers of files in question against it. Update the database after installing new patches or system upgrades.

Check closely the /usr, /usr/bin, /sbin, /usr/sbin, and your X Window binaries directory, because rootkits are usually hiding in these places.

If an attack was successful, a hacker may install a rootkit. This is a suite of applications that can be used for many nasty things (creating back doors, root shells, etc.). It also helps to hide its own presence by modifying system commands that, for example, list all files in the directory (ls, dir (on Linux)) or find any file (find). Therefore, if you suspect that an attacker is on your system, you may not want to trust the ls or find commands, because they most likely have been replaced. How, then, do you list all files in the directory/s to find the rootkit's files?

Alternative Ways to List Files in a Directory

If a rootkit has been installed on your system, it replaced the ls and find commands with trojan versions that will not show a real list, one that includes the rootkit's files.

If you suspect that the current directory may contain a hidden directory or file, do one of the following. If you use Korn shell (ksh), press “ESC=” to list all files in a directory. For example, on Solaris OS:

```
$ ksh -o vi
$ .<ESC>=
1) ../
2) ./
3) .Xauthority
4) .dt/
5) .dtprofile
6) .hushlogin
7) .netrc
8) .rhosts
9) .sh_history
```

Or:

```
$ *
1) TT_DB/    12) mnt/
2) bin/     13) net/
3) cdrom/   14) opt/
4) dev/     15) platform/
5) devices/ 16) proc/
6) etc/     17) sbin/
7) export/  18) tmp/
8) home/    19) usr/
9) kernel/  20) var/
10) lib/    21) vol/
11) lost+found/ 22) xfn/
```

While your ls might be trojaned and not be able to see the hidden files, your Korn shell will see them.

You can also use the echo(1) command, which lists all files in a directory. For example:

```
$ echo *
TT_DB bin cdrom core dev devices downloads etc export home kernel
lib lost+found mnt mynes.txt net nsmail opt patches platform proc proj-
ects sbin test tmp ts1 typescript usr var vol xfn
```

Note: Echo will not show hidden files, such as files starting with “.”.

On Linux systems, you may use the less(1) command to display all files in the directory:

