



Simple Unix Tricks: Detecting Break-Ins

Hal Pomeranz
Deer Run Associates



Who Am I?

- 
- Independent security consultant
 - SANS Institute Senior Faculty
 - Technical Editor for *Sys Admin*
 - Unix Technical Advisor for the Center for Internet Security

Generally speaking, a guy who probably spends way too much time with Unix...



What's In This Course?

- Simple techniques for determining if your Unix system has been broken into
- Uses freely available resources and tools:
 - SANS' "Intrusion Discovery Cheat Sheet"
 - chkrootkit
 - AIDE
- This is *NOT* a course on digital forensics, though some techniques may overlap



What's Your Job?



ASK QUESTIONS!



Simple OS-Level Investigations





SANS' "Cheat Sheet"

- A simple one-page guide to help system administrators look for telltale signs:
 - Strange processes
 - Unexpected files, file modifications
 - Suspicious network usage
 - New cron jobs
 - New accounts
 - Suspicious log entries
- Goal is to use only tools provided with the (Linux) operating system



Important Caveat

- After a root compromise, OS utilities may not be trustworthy due to "rootkit" install
- True forensic investigation is always done with tools brought in from outside:
 - Pre-packaged on CD-ROM
 - Mounted via the network?
- Still, you'd be surprised how many attackers don't bother to cover their tracks in this way



Getting Process Info

- Sometimes simplest is best:

```
ps -ef          # Linux and SYSV
```

```
ps auxww       # BSD
```

- Look for processes you don't recognize
- Helps if you're already familiar with the normal process list for the system
- Also helps if you've already minimized the number of services on the system

More Hints From the Process Table

```
# ps -ef
USER      PID ...  STAT  START      TIME  COMMAND
root      ...    S     Apr15     0:04  init
root      ...    SW    Apr15     0:00  [kflushd]
root      ...    S     Apr15     0:00  gpm -t ps/2
xfs       ...    S     Apr15     0:00  xfs -droppriv -daemon ...
root      ...    S     Apr23     0:00  syslogd -m 0
root      ...    S     Apr23     0:00  klogd
root      ...    S     Apr23     0:00  crond
root      ...    S     Apr23     0:00  inetd
root      1584 ...  S     Apr23     0:00  (nfsiod)
:         :         :         :         :         :
root      ...    S     Apr24     0:00  /sbin/mingetty tty6
root      ...    S     Apr24     0:00  /usr/bin/kdm -nodaemon
root      ...    S     Apr24     0:01  /etc/X11/X -auth /usr/...
root      ...    S     12:33     0:00  -sh
root      ...    R     12:41     0:00  ps -auxww
```

lsof is Also Helpful Here

```
# lsof -p 1584
COMMAND  PID USER  SIZE  NODE NAME
sh       1584 root   4096   123 /dev/.. /lrk5
sh       1584 root   4096    2 /
sh       1584 root 373176 96198 /bin/bash
sh       1584 root 344890 208421 /lib/ld-2.1.2.so
sh       1584 root  15001 208480 /lib/libtermcap.so.2.0.8
sh       1584 root 4118299 208428 /lib/libc-2.1.2.so
sh       1584 root 247348 208459 /lib/libnss_files-2.1.2.so
sh       1584 root 253826 208465 /lib/libnss_nisplus-2.1.2.so
sh       1584 root 372604 208441 /lib/libnsl-2.1.2.so
sh       1584 root 254027 208463 /lib/libnss_nis-2.1.2.so
sh       1584 root   1577   TCP bobo:12497->badguy:1523 (ESTABLISHED)
sh       1584 root   1577   TCP bobo:12497->badguy:1523 (ESTABLISHED)
sh       1584 root   1577   TCP bobo:12497->badguy:1523 (ESTABLISHED)
sh       1584 root   1576   TCP *:12497 (LISTEN)
sh       1584 root   1577   TCP bobo:12497->badguy:1523 (ESTABLISHED)
```



Examining the File System

- If it's Unix, you're going to use `find`:
`find <startdir> <condition> <action>`
- In most cases you'll want to search the entire file system, so `<startdir>` is `"/"`
- The `<action>` is most often `"-print"`
- Let's look at some useful examples...



Wacky File Names

- Find strange file and directory names commonly used by attackers:

```
find / -name ' ' -print
```

```
find / -name '...' -print
```

```
find / -name '.* *' -print
```

- Surprising that attackers continue to use these well-documented directory names...

Set-UID and Set-GID Files

- New or modified set-UID and set-GID files should be a concern:

```
find / \( -perm -4000 -o -perm -2000 \) \  
-type f -ls >setidfiles
```

- Run this command *before* you put the system into production, and save the result
- Audit the system by using `diff` to compare the current output with the saved output



Other Interesting Searches

- 
- Large files (> 10MB):

```
find / -size +10000000c -print
```

- Recently modified files (< 1 week):

```
find / -mtime -7 -print
```

- Not all output is suspicious– run commands regularly to learn what's "normal"



Using the Package Manager

- Software package manager can be used to audit operating system integrity:

```
rpm -Va           # Redhat/Mandrake
```

- Other systems have equivalent functionality (Solaris: **pkgchk**, HP-UX: **swverify**, etc.)
- Assumes attacker hasn't tampered with package management software or database



Suspicious Network Activity

- Check the output of `netstat` and `lsof`:
`netstat -anp` # -p only for Linux
`lsof -i`
- Also check for new entries in `inetd.conf`
- Again, it helps if you're already familiar with what's "normal" for your system
- Eliminating unused network services reduces vulnerabilities and helps auditing



Check for "Promiscuous Mode"

- 
- Network interfaces in "promiscuous mode" means a packet sniffer is running
 - Standard Unix command for checking interface status is `ifconfig`
 - Linux `ifconfig` doesn't accurately report PROMISC mode (use "`ip link`" instead)
 - Solaris `ifconfig` is also broken— use `ifstatus` tool (URL at end of course)



New Cron Jobs

- Look for new `cron` entries, particularly for the root user:
`crontab -u root -l`
- Probably should also check the integrity of the `cron` daemon itself:
 - Via the OS package manager
 - Comparing MD5 checksum from other system
 - Against vendor checksum database



Suspicious Accounts

- Look for extra UID 0 accounts:

```
awk -F: '($3 == 0) { print $1 }' /etc/passwd
```

- Accounts with no password set:

```
logins -p          # not available on all Unix systems
```

```
awk -F: '($2 == "") { print $1 }' /etc/shadow
```

- May also want to check that "system" accounts are still "blocked"



Check Your Logs!

- 
- Failed logins and failed su attempts
 - Network connections from unknown or suspicious network ranges
 - Interfaces go into promiscuous mode (Linux)
 - Strange messages from RPC-based services with lots of non-printable characters
 - Bizarre or long addresses in Sendmail logs
 - Large numbers of errors in web server logs



Additional Utilities: chkrootkit





What is It?

- 
- A simple shell script that looks for "signatures" of common rootkits
 - Comes with some helper programs with more advanced capabilities
 - Able to detect even some kernel rootkits
 - Ported to many Unix variants, but clearly designed primarily for Linux and FreeBSD



Same Problem Again

- As with manual investigation, chkrootkit relies on certain shell utilities
- Attacker may have replaced OS utilities with Trojan-ed versions to spoof admin
- chkrootkit options:
 - Alternate \$PATH: `chkrootkit -p <dir>:...`
 - Alternate mount: `chkrootkit -r /mnt`



Simple chkrootkit Checks (1)

- chkrootkit first runs `strings` on several dozen OS binaries
- Looks for strings that are present in known Trojan versions
- Obviously will not recognize Trojans that have not yet been discovered/categorized
- "Expert mode" (`chkrootkit -x`) shows full `strings` output for admin review



Simple chkrootkit Checks (2)

- `chkrootkit` looks for files or file changes created by well-known rootkits
 - "aliens" check covers many signatures
 - Specific functions for other rootkits
- `chkrootkit -l` lists available checks
- Select individual checks on command line (default is to run all checks):
`chkrootkit aliens scalper slapper`



"bindshell" Check

- 
- Compares the output of "`netstat -an`" against a list of common back-door ports
 - False-positives are common due to:
 - Hosts running Portsentry/Klaxon/Wrappers
 - Local services listening on odd ports
 - Again, know thy systems!



Looking for Kernel Rootkits

- 
- Some kernel-level rootkits show up due to strings found in `/proc/ksyms`
 - May be able to find hidden processes by exhaustive traversal of `/proc`
 - Possibly detect hidden directories due to parent directory link count discrepancies



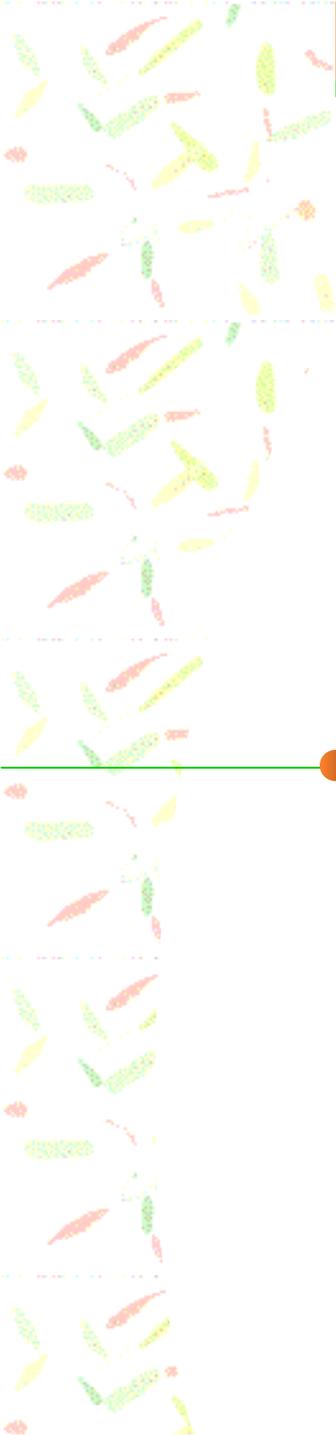
Groveling Through `/proc`

- Kernel rootkits hide `/proc/ <pid>` dirs in normal listing, but directories still "exist"
- Trivial algorithm (`chkproc`):
 - First get directory listing from `/proc`
 - Now run through entire PID range, attempting to open `/proc/ <pid>`
- Can generate false positives when processes started during `chkproc` run



Directory Link Counts

- The link count on a directory should be two plus the number of sub-directories:
 - Count normal directory entry plus "." link
 - Each subdir has ".." link that points to parent
- Kernel rootkits often "hide" a directory but forget to reduce parent directory link count
- `chkdirs` program walks entire file system looking for link count discrepancies
- Not part of standard checks– run manually



Additional Utilities: AIDE





How It Works – Overview

- 
- Create config file listing critical files and directories to watch
 - Generate initial file/checksum database for this list of files
 - Periodically re-run AIDE to compare current file/directory info to database
 - Report discrepancies



What Problem Does It Solve?

- 
- Lets you know *exactly* which files have been changed on your system
 - This is indispensable information after a security incident
 - However, the greatest recurring value may be alerting you to mistakes by local admins

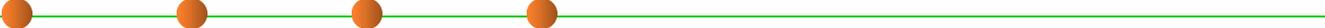


The Problem

- 
- An attacker who roots your box can modify your AIDE binary/database
 - Solutions include:
 - Binary and database on CD-ROM
 - Read-only NFS from central, protected host
 - Remote checks via SSH from central host
 - Read-write local access with periodic external verification



What About Tripwire?

- 
- Tripwire was the first integrity checking tool for Unix systems
 - Originally a grad student project by Gene Kim, and distributed freely
 - Tripwire is now a commercial product
 - Older version for Linux was released under the GPL, also ported to FreeBSD

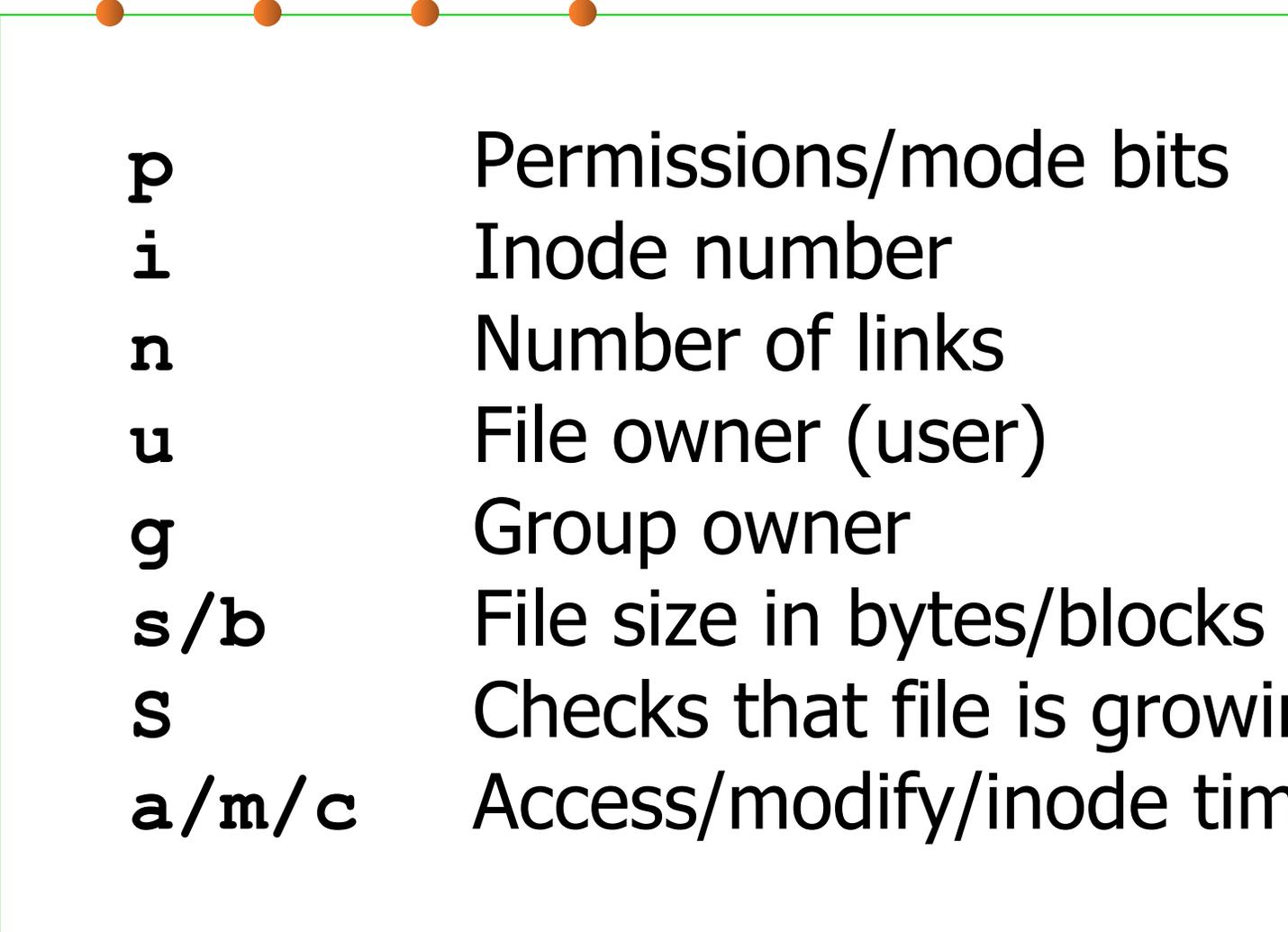


AIDE Installation Notes

- 
- Includes standard "configure" script
 - However, insists on you already having a number of other Open Source tools:
 - GNU `bison`, `flex`, and `make`
 - Zlib data compression library
 - mhash library (checksum algorithms)
 - Source tweaks may be required for non-mainstream operating systems



aide.conf – Per File Checks



p	Permissions/mode bits
i	Inode number
n	Number of links
u	File owner (user)
g	Group owner
s/b	File size in bytes/blocks
S	Checks that file is growing
a/m/c	Access/modify/inode timestamps



aide.conf – Checksums

- 
- Checksums include md5, sha1, tiger, rmd160, haval, gost, and crc32
 - Use multiple checksums on "critical" files for maximum security
 - Use single checksum on normal files to reduce system impact of audit

aide.conf – File Entries

- Specify file regexp and list of parameters:

```
/usr/bin/su$    p+i+n+u+g+s+m+c+md5
```

- Common sets have pre-defined macros:

```
R    p+i+n+u+g+s+m+c+md5 ("read-only")
```

```
L    p+i+n+u+g    ("log file")
```

```
>    p+i+n+u+g+S  ("growing log file")
```

```
E    Empty set ("ignore everything")
```



aide.conf – Directories

- By default, AIDE recursively descends through directory trees, catching all entries

- Use `!/= to modify this behavior:`

```
=/usr$           R  # check /usr itself,  
                  # but don't recurse
```

```
/etc/namedb     R  # watch zone files  
!/etc/namedb/slave  # but not slave files
```

Partial aide.conf File

```
database=file:/var/aide/aide.db                #where DB lives
database_out=file:/var/aide/aide.db.new        #put new DB here
verbose=20                                     # 0-255
H = p+i+n+u+g+s+b+m+c+md5+sha1+rmd160       #"heavy" auditing

/dev                                           L                #watch /dev entries
!/dev/[pt]typ[0-9a-f]$                        #these change a lot

/root                                         H                #critical area
/root/.ssh/known_hosts$                       >                #this file changes

=/etc$                                        L                #critical directory
/etc/*.                                       R+sha1           #watch contents
!/etc/ntp.drift$                              #ignore this file
```



Files/Directories to Watch

- 
- "Significant" directories like `/`, `/usr`, `/var`, `/dev`, `/tmp`, and `/var/tmp`
 - Dot files in root's home directory (but beware files generated by SSH)
 - `/etc` (but beware derived files in `/etc`)
 - Crontab files and directories
 - Kernel and boot loader (if any)



Also Watch `bin` & `lib` Dirs

- Monitor *all* `bin` and `lib` dirs on the system (including `/opt` and `/usr/local`)
- Again, use single checksum except on "critical" files to improve scan speed
- "Critical" files include:
 - System shells (`sh`, `csch`, `ksh`, `bash`, ...)
 - Daemons (`inetd`, `syslogd`, `sshd`, ...)
 - Authentication (`login`, `su`, `passwd`, ...)
 - Forensic tools (`ls`, `ps`, `netstat`, `ifconfig`, ...)



Don't Forget "Content" Dirs!

- 
- Web server doc trees and CGI bins
 - Anonymous FTP areas
 - DNS zone files
 - NIS maps (if not in kept /etc)

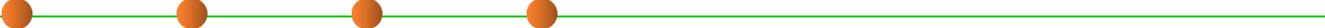


The Problem With Log Files

- 
- Monitoring log files might seem like an obviously good idea
 - The problem is that log files get moved, "rotated", and archived
 - Generally, it's only a good idea to watch stationary log files like `utmp/wtmp`



Using AIDE



■ Generating your database:

```
# aide --config=/var/aide/aide.conf --init  
  [... some informational messages not shown ...]  
# mv /var/aide/aide.db.new /var/aide/aide.db
```

■ Running a check:

```
# aide --config=/var/aide/aide.conf --check
```

```
AIDE, version 0.10
```

```
### All files match AIDE database.  Looks okay!
```

Aide Reports a Change...

```
# /var/aide/aide --config=/var/aide/aide.conf
AIDE found differences between database and file system!!
Start timestamp: 2004-03-21 16:14:28
Summary:
Total number of files=20396,added files=0,removed files=0,changed..

Changed files:
changed:/etc/mail/statistics
changed:/etc/security/audit_data
Detailed information about changes:

File: /etc/mail/statistics
Mtime      : 2004-03-21 13:47:55      , 2004-03-21 16:02:57
Ctime      : 2004-03-21 13:47:55      , 2004-03-21 16:02:57
MD5        : Vhbdo2DxMxuwRZJE9+610A== , rc5K7XRiUfKJ0cET3jATYg==
SHA1       : 8JkRx12+8u6/RrxevzPraG... , 020pi+SSSmAej/PraA/vwgJa...

File: /etc/security/audit_data
Mtime      : 2004-03-21 13:00:00      , 2004-03-21 16:00:00
Ctime      : 2004-03-21 13:00:00      , 2004-03-21 16:00:00
[...]
```



Thoughts on Automation

- 
- You want to run AIDE from `cron`
 - You don't want to get spammed if everything is OK
 - Simple script (next slide) can differentiate normal output from real warnings
 - May want to run periodic manual audits just to make sure things are working



Here's That Script...



```
#!/bin/sh
```

```
TEMPFILE=/var/aide/.out$$
```

```
/usr/local/bin/aide --config=/var/aide/aide.conf \  
>& $TEMPFILE
```

```
if [ ! "`grep '### All files match' $TEMPFILE`" ]  
then
```

```
    cat $TEMPFILE
```

```
fi
```

```
rm $TEMPFILE
```



Updating Databases

- 
- Files will change during the lifetime of a system and database must be updated
 - Use `"aide --update"` to run a scan and simultaneously produce new database
 - Be sure to carefully check scan report before overwriting old database!



Wrap Up





That's All Folks!

- 
- Any final questions/comments?
 - Please fill out your eval forms!
 - Thanks for listening!

Plenty of useful URLs to follow...



Misc References

- SANS "Intrusion Discovery Cheat Sheet":
http://www.sans.org/score/checklists/ID_Linux.pdf
- Chkrootkit home page (good links!):
<http://www.chkrootkit.com>
- SANS "Reading Room"
<http://www.sans.org/rr/>
- CERT/CC "Tech Tips"
http://www.cert.org/tech_tips/



AIDE Info

- Homepage (w/ docs), download site:
<http://www.cs.tut.fi/~rammer/aide.html>
<http://sourceforge.net/projects/aide>
- Sample config files:
<http://www.deer-run.com/~hal/aide/>
- Additional software needed:
GNU Software – *<http://www.gnu.org/>*
Zlib – *<ftp://ftp.info-zip.org/pub/infozip/zlib/>*
Mhash – *<http://mhash.sourceforge.net/>*



Other Software



- **ifstatus**

<ftp://ftp.cerias.purdue.edu/pub/tools/unix/sysutils/ifstatus>

- **lsof**

<ftp://vic.cc.purdue.edu/pub/tools/unix/lsof/>

- **Tripwire (commercial version)**

<http://www.tripwire.com>

- **Tripwire (Open Source for Linux/FreeBSD)**

<http://www.tripwire.org>