How to (hopefully) avoid getting r00ted.

Stephen Quinney <squinney@inf.ed.ac.uk>

Basis of the Talk

- Last year we got r00ted.
- We wrote up what happened.
- We also did an extensive review.

- http://www.dice.inf.ed.ac.uk/publications/2011/ssh_compromise_report.pdf

Main Topics

- Threats
- Risks
- Defence
- Monitoring
- Auditing

Threats

- Advanced Persistent Threat
  - An attacker who - for whatever reason - wants to attack you. Against this sort of attacker, the absolute level of your security is what's important.
- Random Attacks
  - Security against this sort of attacker is relative; if you're more secure than almost everyone else, the attackers will go after other people.

http://www.schneier.com/blog/archives/2011/11/advanced_persis.html

Risks

- Remotely exploitable security hole.
- Misconfiguration.
- Attacker acquires user credentials and uses local exploit.

Defence

- Minimise the attack surface.
- Strength in depth.
- Keep it simple.
- Design to mitigate the risk of misconfiguration.
- Monitor login failures & respond appropriately.
- Make the attacker's life difficult.

Minimise the Attack Surface

- Only allow access to the smallest possible set of hosts.
- Only allow access to the smallest possible set of ports.
- Only run the smallest possible set of services.

openssh configuration

- Never allow root logins
- Limit authentication types.
- Restrict access to authorized groups/users.

Love your PAM stack!

- Horrible syntax but...
- Essential for creating a secure system.
- Provides a huge variety of ways to control access and monitor user activity.
- Be aware: some authentication processes do NOT use the PAM auth stack, e.g.
  - SSH using public key
  - SSH using GSSAPI

pam_access

- Rules in `access.conf` file.
- Control access based on:
  - username
  - user in group (or netgroup)
  - origin (host or domain names and addresses)
- First matching entry applies.
- Can permit and deny access.

Handling Brute Force Attacks

- fail2ban - http://www.fail2ban.org/
- denyhosts - http://denyhosts.sourceforge.net/
- SSHGuard - http://www.sshguard.net/
- pam_tally2
- pam_abl - http://pam-abl.deksai.com/

fail2ban

- Monitors log files for signatures of login failures.
- Responds to multiple failures from a single IP within a certain time period.
- Can configure firewalls, tcpwrappers, etc.
- Blocks are dropped after a certain time period.

pam_tally2

- Tracks failed logins.
- Denies access once a limit has been exceeded.
- Denies access for a defined period of time.
- Command line tool for querying and modifying entries in the failure count file.

Anatomy of an Attack

- Aim to collect “sensitive” data.
- Uses a “rootkit”:
  - Hides itself from simple searching.
  - Modifies log files.
  - Keylogger.
  - Backdoor.
  - Communication channels.
  - May replace core binaries.
  - May use “trojans”, e.g. SSH daemon.

Making the Attacker's Life Difficult

- Install the smallest possible set of software.
  - Particularly try to avoid providing compilers.
- Block loading of kernel modules.
- File-system partitioning to restrict suid scripts.
- Store logs centrally.
- Disable old accounts.
- Block some outgoing traffic.

Disable Kernel Module Loading

- Most simple and effective way of blocking the installation of rootkits.
- Irreversible except by reboot.
- Consider doing this immediately after system boot has completed.

```
echo 1 > /proc/sys/kernel/modules_disabled
```

Monitoring

- Do NOT wait until it is too late!
- Do regular sweeps (e.g. daily).
- Various tools available:
  - chkrootkit - http://www.chkrootkit.org/
  - Rootkit Hunter (rkhunter) - http://www.rootkit.nl/projects/rootkit_hunter.html
- Consider running more than one.
- Do not rely on the system tools.

Auditing

- 'last' log is not sufficient.
- Enable process accounting.
- Use the Linux audit system.

Linux Audit System

- Not just for SELinux.
- Records user logins.
- Can watch syscalls.
- Can watch for filesystem access.
- Can record events from any trusted app.
- Can securely log to a remote host.

Auditd – Watching the Filesystem

```
-w /bin   -p wa -k FS_mod
-w /boot  -p wa -k FS_mod
-w /etc   -p wa -k FS_mod
-w /lib   -p wa -k FS_mod
-w /lib64 -p wa -k FS_mod
-w /sbin  -p wa -k FS_mod
-w /usr   -p wa -k FS_mod
```

Auditd – Watching setuid Files

```
-w /bin/mount          -p x -k FS_suid
-w /bin/su             -p x -k FS_suid
-w /bin/umount         -p x -k FS_suid
-w /usr/bin/chage      -p x -k FS_suid
-w /usr/bin/chfn       -p x -k FS_suid
-w /usr/bin/chsh       -p x -k FS_suid
-w /usr/bin/gpasswd -p x -k FS_suid
-w /usr/bin/ksu        -p x -k FS_suid
-w /usr/bin/newgrp     -p x -k FS_suid
-w /usr/bin/passwd     -p x -k FS_suid
-w /usr/bin/pkexec     -p x -k FS_suid
-w /usr/bin/sudo       -p x -k FS_suid
```

Auditd – Watching syscalls

```
-a always,exit -F arch=b32 -S init_module -S
delete_module -k modules
```

pam_loginuid

- Sets a loginuid attribute for the authenticated process.
- Used for auditing.
- Cannot be altered (unlike uid, euid).
- Attached to all child processes.

pam_tty_audit

- One for the truly paranoid.
- Used to audit **all** activity on a tty.
- Inherited by all child processes.

- For example, audit all root activity:

```
session required pam_tty_audit.so \
                  disable=* enable=root
```

Auditd – reports & searches

- aulast - 'last' like
- aulastlog - 'lastlog' like
- aureport - Summary reports
- ausearch - Complex searches
- audit-viewer – graphical tool

Summary

- Don't wait until you've been r00ted to start thinking about security!
- Understand the threats and risks.
- Consider the strength of your defences.
- Monitor your systems frequently.
- Prepare a plan for what to do when the worst case happens.