Real-World Linux Security

Rick Moen, rick@linuxmafia.com

No-bafflegab zone

This is a real-world overview for everyone.

- Plain English
- No security-biz bullshit.

'Security' just means making usage match authorisation.

Hi, I'm a happy new Linux user!

Reg. user: KA-POW! Can shoot yourself in the foot.

Root user: KA-POW! Can shoot everyone and everything in the foot.

(sudo, as normally used: See root user.)

Groups you're in and 'world' have some shooting rights.

Process running as user FOO can shoot

exactly as FOO can.

Hi, I'm a happy new Linux user!

Day-to-day activity involves processes, rights, ownership.

- Processes mutter to STDOUT and logfiles. Troubleshooting starts there.

- Some processes run with more 'privilege' than others, hence more security-sensitive.

But also....

Software entomology

News flash: Software has bugs.

- The classic 'oops!' or
- 'People shouldn't *use* my code that way.'

Most coders are optimists.

Bugs get exploited.

Fortunately, your user's processes can do limited damage (deliberately). Don't fool with this.

Worst threat against your system: YOU.

The Things Security Covers

- Prevent and Detect – almost all you ever hear about

But also:

- Limit exposure
- Defend in depth
- Harden
- Identify attackers (rarely)
- Recover – arguably the most vital (do first!)

Rebuild and Backup/Restore

- If you can rebuild your OS, and restore your configs and data files, then you can mostly relax.
- It's a 'restore system', not a 'backup system'.
- Example details on a page at linuxmafia.com.

Testing!

- The only way to know your plans for rebuilding and restoring will work is to *try them*.
- Don't use Big Dumb Copy methods. Rely on your distro package system. E.g., if you're backing up /usr, you're doing it wrong.
- Simpler is better.

Distro Maintainers: Your Friends

- Stick with an actively maintained distro release.
- Read its security-alerts mailing list.
- Use distro package maintenance.

No, Really. They're Your Friends.

- You're in the big leagues. Think ten times before downloading www.crooked.com/runme and doing 'sudo sh /tmp/runme' or similar.
- Software & other packages from outside the distro regime are *very* suspect.

Unavoidable Exceptions

Sometimes you cannot avoid non-distro origins.
Drawbacks:

- Distro maintenance cannot update it and doesn't even know it's there.
- You won't get updates unless you take steps.
- It becomes your job to vet / inspect downloads.
- You don't get packager gatekeeping.

When you must install 'local' software

- Best choice: Package in a reputable third-party repo.
- Next: Package in a reputable outsider's repo.
- Next: Compile from tarball. (Distro-wrap, if possible.)
- Worst: Download a binary.

Examples of trojan/fake tarballs, gnome-look.org phony.

The problem of maintenance is why many recent attempts to threaten Linux have aimed at buggy Web apps locally installed to some Web servers.

Server problems are out of this talk's scope, but buggy PHP apps & configurations are legion.

Occasionally check to see if there's a package.

Sucker-punch software

- Adobe Flash
- Adobe Acrobat Reader (esp. its Javascript)
- Oracle / Sun Java

Frankly, these are just not safe for handling arbitrary content off the Internet.

Evince/xpdf for Internet PDFs. (Windows? Try muPDF.)
Flashblock or NoScript mitigates Flash risks.

How to Recognise Hopeless Software

- Basically the same thing keeps getting fixed ('patched') over and over.
- Especially if the recurring problem is input validation.

Example: lpd, the old printing daemon, thankfully replaced by CUPS.

How to Read Security Advisories

Skim-read the mailing list.

- If it's software you omit, hit 'next'.
- If it's software you installed but weird configs only, hit 'next'.
- If it says 'could make the software crash/hang', hit 'next'.
- If it says '*potentially* execute arbitary code', it's usually a vague handwave and you needn't hurry.
- It says 'privilege escalation', worry a bit more.
- If it says 'buffer overflow', worry on *privileged* code.

Most fixes in Unix are anticipatory.
Many fixes close holes that *might*, maybe, someday be exploitable. Perhaps. With a strong tailwind.

How to Read Security Advisories

These are not threats against *you*:

- Cross-site scripting (XSS) problems.
- Denial of service (a problem, but not a break-in).
- Rootkits. (Muddy tracks left behind by the burglar.)

What about Hardening, Defence in Depth, etc.?

- Caution: Beware overcomplexity when tempted to 'add security stuff'. Lesson of portsentry/fail2ban, etc
- Yes, iptables rules (packet filters) and checkers like IDSes are 'defence in depth', but apply the KISS principle. Remember what the worst threat is.

That having been said, logwatch/logcheck and IDSes like OSSEC bear consideration (but require tuning).

All Sorts of Further Resources

- Marcus Ranum on The Six Dumbest Ideas in Computer Security http://www.ranum.com/security/computer_security/editorials/dumb/
- Why local packages are to be avoided where feasible: http://linuxmafia.com/~rick/weatherwax.html#1
- Marcus Ranum's rant 'What Sun Tsu Would Say”: http://www.ranum.com/security/computer_security/editorials/master-tzu/
- Open Web Application Security Project (OWASP) https://owasp.org/ Skim-read the site for basic principles; beware of gadget-freakery.
- Firewalls and Internet Security, 1st Edition (online), http://www.wilyhacker.com/1e/ Utterly excellent for learning principles.

A Word from Our Sponsor

This talk sucked! You can do better. (I wrote this presentation in three days, after our originally scheduled speaker had to cancel.)

SVLUG needs your presentation. You know you want to!