Server Security in Practice

Geoffrey Thomas <geofft@mit.edu>

SIPB Cluedumps 2011

Our Environment

- SIPB: volunteer student computing organization
- Runs services for the MIT community
- Uses MIT/Athena infrastructure
- Volunteer-run, mostly on student schedules
- Relatively low-budget

scripts.mit.edu

- Web server platform using Athena infrastructure
- 4000+ accounts, 900+ .mit.edu names
- Five web VMs, 2 MySQL servers, 2 load-balancers
- Supports all common and many uncommon scripting languages and frameworks
- Shell access, cronjobs, procmail, etc.

Basic security steps

- No remote password logins, only Kerberos or SSH keys
  - Passwords work locally (including VM consoles)
- No access from personal accounts, either via Kerberos or sudo/su
- Only log in as root from secured, personal machines

Basic security steps

- Watch server logs for suspicious behavior, and respond quickly
- Install only software from trusted sources
- Package our own software and version-control our own configuration
- Regular software updates (yum-updatesd, apticron)

Web server security

- Privilege separation: suEXEC
  - Code is only readable to the account in question
  - Any problem is contained within the account
  - Run everything through suEXEC: static-cat
- Limit compromised accounts
  - Scan for web application updates
  - Automatic web application updates

Clean development practices

- Separate Kerberos principal required, two-factor authentication recommended
- All code is open, so development can be done unprivileged
    - Kerckhoff's principle (no security through obscurity)
- VCS repository is considered unprivileged
- Be careful at the boundary between unprivileged and privileged/trusted

XVM

- Virtual machine hosting service
- Eight hosts, 900 virtual machines
- Click a button on a web form and get a server
- Full ability to customize your virtual machine

Security model

- Everything uses Kerberos, including server-to-server auth
- Separation of privilege vs. separation of VMs
- Defined interfaces (remctl)
- Anything user-editable we don't touch

Linerva

- University-wide Linux shell access computer
- 500+ users currently logged on, 2500+ unique users in the last year
- Allows password authentication, and web logons via https://linerva.mit.edu/
- Lots of popular software installed; will install software on request

Securing a public-access Linux server

- UNIX privilege separation rather good
- Exception: setuid
- Linux kernel requires frequent security updates
    - Ksplice
    - Require mmap_min_addr > 0
    - Disable module autoloading
- Check new packages for daemons

Security and responsiveness

- You cannot delay critical security updates
- Design your system so updates are easy
    - Replication / redundancy
    - “Document” state, so you are not afraid to make changes
    - Keep up with OS updates
- Pay attention to security issues: security announcement lists, LWN, etc.

Principle of least privilege

- A successful attack compromises as little as possible
- If your stuff gets broken into,
  - our stuff doesn't get broken into
  - other peoples' stuff doesn't get broken into
- UNIX accounts and permissions are effective, but subtle
- Virtualization very effective, and cheap/easy

Least privilege and scripts.mit.edu

- No password login for anyone; log in from Athena (and don't forward Kerberos tickets)
- All access is through separate AFS identity “daemon.scripts” in global credentials
- Kernel hacks to partition usage of global credentials to a user's own AFS volume
- Even root/httpd cannot read users' files

Least privilege for user identities

- Both scripts.mit.edu and Athena have a concept of “local” and “non-local” users
- Non-local users come from a somewhat less trusted source (Hesiod, LDAP)
- Account spoofing attack: claim that user “geofft” has uid 0
- Custom NSS module, nss_nonlocal
- Prevents server compromise from spreading

Principles for system design

- “The enemy knows the system”
- Clean development practices
- Administrative access must be secure
- Security updates in hours, not days
- Use privilege separation wherever possible
- Don't be afraid to design new approaches
    - But understand the existing ones
- Your system will be better for users