Security - Linux Security (part 2)

Note Title 3/31/2011

Announcements

- Quiz today!
- In-class review of paper today (so final version due Tuesday, along with today's review)
- Lab 4 is up; due on Wednesday, April 13 (Checkpoint next Wednesday)
- No 2nd midterm in this class; Final exam is Thursday, May 12, @ noon

Recap of System Hardening:

- Limit applications
- Patch management
  (0-day vulnerabilities)
- TCP Wrappers
  - before allowing any tcp connection to any service, tcpd first evaluates a list of access controls in /etc/hosts.allow + /etc/hosts.deny

    (a bit obselete + less powerful than iptables)

- Fire walls - iptables
  - flexible & very powerful

- Antivirus Software
  (more worms than viruses, historically)
  - some freeware, eg ClamAV
  - many comercial coming out: McAfee, Symantec, Sophos

- User/Password management

- Logging →

Logging

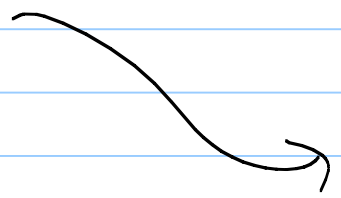
- Generally run by syslogd
- Better choice is Syslog-NG
  - can use much wider variety of sources + destinations
  - rules engine is much more flexible
  - supports logging via TCP (which can be encrypted)

Other Tools worth mentioning

- Bastille - a system hardening utility (educates as it secures)
- Tripwire - utility that maintains a database of critical files & changes to them Shadow file for passwords
- Snort - powerful (free!) IDS
- Nessus - security scanner

Application Security

- Minimize processes running as root
- Modularity: Postfix versus Sendmail, new Apache server distributions
- Encryption
- Logging
- Chroot jail



Chroot 'jail'

If a process is only writing to one directory, eg. /srv/ftp/public, then the daemon should not have access to anything else in the filesystem.

So we'll "map" this directory to look like root, /, so that the daemon can't see anything else.

(Adds complexity, but increases security.)

Mandatory Access Controls in Linux

As we've said, Linux is based on DAC.

MAC is safer; consider root.

- In MAC setting, root is used to administer security policy only.
- System administation is done with other accounts that can actually change the system.
- No "root takes all" issue & root is less commonly used.

SELinux

- NSA's implementation of MAC for Linux
- Doesn't change basic DAC in Linux, but adds MAC on top.

Subjects and objects

Subjects = processes (not users)

objects fall into categories, such as:
- file
- dir
- socket
- node
- xserver
- ...

- Each category of object has a set of possible permissions:
  - search
  - rmdir
  - getattr
  - reparent

} dir

Rule Structure:
1. that which is not expressly permitted is denied
2. allows subjects, permissions and objects to be grouped.

Every individual subject & object controled by SELinux gets a security context, which is (user, role, domain)

not Linux users - separate authentication

"sandbox"

admin add to files

#1 key to SELinux

(called Type Enforcement)

Two types of decisions for SELinux:

1. Access decisions
   can you read/write, etc. to files, etc.

2. Transition decisions
   Sometimes, you'll need to create a process or file for a different "sandbox".

Other models in SELinux are possible.

- RBAC can be added.
- Multilevel security - specifically, Bell-LaPadula Model
  (enforced via file system labeling)

These SE Linux policies are actually specified in a bunch of text files in /etc/security/selinux.

Also GUI-based admin tools:
- http://www.tresys.com
- in Red Hat & Fedora, use
  → system-config-securitylevel

- Wicked learning curve
  (take extensive training & experience)

Novell AppArmor

- Much easier to administer
- Only available for SuSE Linux (+ Ubuntu)
- More limited:
  - primary goal is to restrict applications
  - only works for a subset of applications
  - No RBAC or multi level security

(root is still root, unless the application is covered)