Security principles

Host security

Host Security: Outline

- General Security Principles
- How to detect Break-ins
- How to avoid Break-ins
- Tools to Protect Your System
- Host Security Checklist
- Additional Security

Security Principles

- Defense In Depth: placing security mechanisms in place at many different levels in your system architecture
- It means doing security in your network
  - on your operating systems
  - on your applications
  - at all levels where possible.

Security Principles

Risk Analysis: identifying the things within your organization that are valuable and should be protected.

If you have something extremely valuable...

- Do you really want to connect it to the Internet?
- What amount of security is needed to protect it?

Security Principles

Security Policy: the set of rules that you put in place to make sure your systems stay secure.

Having these policies makes it easier to consistently implement good security practices.

Security Policy Examples

- Identity Management: rules about who gets accounts, what they can access, how long they get to keep the accounts
- Security Incident: What constitutes an incident, responsibilities of each party, who gets notified and when, legal implications, etc.

Security Principles

End-to-End Security, in contrast to host security, is security which is implemented at the application level. In this case, even the host itself need not be trusted, since the application itself supplies the security.

For example, PGP encrypted e-mail is encrypted by the sender within the e-mail application, and decrypted by the recipient. The message is protected from one end to the other.

Security Principles

Integrity – filesystem integrity is the ability to detect authorized and unauthorized changes in the filesystem.

ex. a hacker breaks into your system, and adds a new username to the /etc/passwd file with a uid of 0. How would you know that this has happened, unless you were watching for file changes?

Host Security

Host Security refers to securing the operating system, filesystem and the resources of the Host from unauthorized access or modification or destruction.

This is in contrast to things like: firewalls and VPNs (network security) or Apache or Oracle penetration testing (application security).

Doing a good job at Host Security on all of your hosts is one of the most important ways to prevent breakins.

Vulnerabilities

Common Vulnerabilities are the areas where you are most likely to get break-ins. These are areas to focus on:

- Public Services: FTPD, Apache, MySQL, PHP
- User Accounts: default accounts, weak passwords, automatic logins
- Old Software: BIND/named, etc.
- Phishing Attacks via E-mail and the Web

Break-ins

What does a breakin look like?

- /var/log directory wiped out
- /var/log/wtmp wiped out
- /sbin/init (ps, lsof, netstat) replaced with a new file
- New SUID root programs
- Daemons listening on Unknown network ports
- Users logged in from remote locations
- Large amounts of network traffic-- SMTP, IRC, HTTP

Break-ins: Avoidance

So now that we know what to watch out for, we have a list of things we need to do:

- See what services are running
- See what network ports are listening
- See if any files have been changed
- Manage user accounts
- Monitor who logs into the system
- Keep system software is up to date

Host Security Tools

- PROCESSES: ps, top, acct
- OPEN FILES: lsof
- CHANGED FILES: incrond, debsums, fcheck
- NETWORK: netstat, nmap
- SUID: find, /etc/sudoers
- LOGINS: last, lastcomm, sa, acct
- UPDATES: apt
- SERVICES: services, update-rc.d

Tools: nmap

- One of the oldest network port scanners
  - see: http://nmap.org/
- More than just a “port scanner”, it is also able to run scripts to scan for additional software vulnerabilities
- Available with a GUI it is called “zenmap”

Tools: nmap

- TCP SYN Scan -- nmap sends SYN, but does not complete the connection, “TCP Half-Open”
- TCP CONNECT Scan -- nmap, TCP full connection
- TCP Bad Flags Scan -- nmap uses odd or illegal TCP flag combinations to produce error responses
- TCP ACK Scan -- nmap sends ACK to produce error response
- UDP Scan -- nmap sends a UDP packet

Tools: nmap examples

- nmap <SCANTYPE> <PORTS> <HOSTS>
- nmap -sT -p80 pcXX # TCP Port 80
- nmap -sS -p U:53,137,T:21-25,80 pcXX # Multiple Ports
- nmap -sT -p80 196.200.216.0/24 # A Whole Subnet
- nmap -A -T4 pcXX # OS FingerPrint
- nmap -v -sn 192.168.0.0/16 10.0.0.0/8 # Ping Subnets

Tools: nmap

- NSE, the Nmap Scripting Engine
- Can run complex tests against services
- ex. nmap -sC pcXX
- ex. nmap –script smb-os-discovery somepc

Tools: acct

Records process termination records for all processes in the system. (see “man acct 5”)

- records are stored in /var/log/account/pacct
- sa command and lastcomm commands are used to summarize the output of these records
- ex. % sudo sa -u
- ex. % sudo lastcomm sysadm
- NOTE: this set of tools is not installed by default
- NOTE: remember, records are written on command termination, so **after** the command has finished

Tools: fcheck

fcheck – stands for filesystem integrity checker

- generates md5sums of entire directories of files
- these md5sums are updated periodically, and the change log is mailed to the root user
- % sudo apt-get install fcheck
- % sudo fcheck -ac # build initial database
- % sudo fcheck -a # run a scan
- can be very cpu intensive on some systems
- configuration is in /etc/fcheck/fcheck.cfg

Tools: fcheck

One note about periodic filesystem intergrity checking: this method will only produce notifications at the next time the test is run.

In many cases, during the breakin, the hacker will wipe out your logs, and break your monitoring systems. It is much better if you could be notified immediately when a change happens, i.e. in real-time.

Tools: incrond

- incrond: the inotify cron daemon
- a real-time kernel feature that supports logging when there are any changes to any inode in the filesystem
- allows you to construct “watch lists” of all of the directories or files you want to monitor for changes
- can send log messages or run scripts upon any/all operations on files: open, create, modify, etc.
- inotify is the default in the kernel, but the incrond package is not installed by default
- real-time notification makes this extremely valuable for detecting breakins

Tools: incrond example: /etc/incron.d/globals

- /etc IN_MODIFY,IN_CLOSE_WRITE,IN_CREATE,IN_DELETE /usr/bin/logger -p news.warn "$% $@/$#"
- /sbin IN_MODIFY,IN_CLOSE_WRITE,IN_CREATE,IN_DELETE /usr/bin/logger -p news.warn "$% $@/$#"
- /usr/sbin IN_MODIFY,IN_CLOSE_WRITE,IN_CREATE,IN_DELETE /usr/bin/logger -p news.warn "$% $@/$#"
- /bin IN_MODIFY,IN_CLOSE_WRITE,IN_CREATE,IN_DELETE /usr/bin/logger -p news.warn "$% $@/$#"
- /usr/bin IN_MODIFY,IN_CLOSE_WRITE,IN_CREATE,IN_DELETE /usr/bin/logger -p news.warn "$% $@/$#"
- /usr/local/bin IN_MODIFY,IN_CLOSE_WRITE,IN_CREATE,IN_DELETE /usr/bin/logger -p news.warn "$% $@/$#"
- /usr/local/sbin IN_MODIFY,IN_CLOSE_WRITE,IN_CREATE,IN_DELETE /usr/bin/logger -p news.warn "$% $@/$#"

Tools: debsums

- tool used to create md5sums of standard distributions of debian packages
- % sudo debsums_init is used to create initial checksums
- can be run from the command line to check the integrity of system files: % debsums -ce
- can also check the integrity of system config files, but that tends to be less useful
- part of /etc/apt/apt.conf.d/ daily package runs
- once again, not installed by default

Tools: service

A shortcut command for accessing the /etc/init.d/<name> scripts. Used for 1) starting services, 2) stopping services, 3) checking status, 4) enabling or disabling services for machine startup... [see also: initctl]

- sudo service apache stop
- sudo service apache start
- sudo service apache status
- sudo update-rc.d apache enable

Checklist: disable services

/etc/init.d scripts

% sudo initctl list | more

% sudo service <name> start/stop/restart
% sudo /etc/init.d/<name> stop/start/restart
% sudo update-rc.d <name> disable/enable

Host Security

Now we know that there are easily available tools that can provide good host security, what's the next step?

We need to use these tools every time we install a new system, and continue to use them to monitor our systems over time.

Let's make a checklist so that we apply these tools in a consistent and easy way every

Host Security: checklist

We're going to do this on every system we own, every time we do an installation.

1. Apply System Updates
2. Enable Automatic Updates
3. Turn Off Unnecessary Services
4. Enable System Accounting
5. Enable Inotify Logging
6. Enable Filesystem Checksums

Host Security: Additional Topics

- chroot: restricting applications to subdirectories
- AppArmor: role-based application protection
- SELinux: comprehensive system access control
- Firewalls: iptables using ufw

We can start to talk about any/all of these after the exercises.