Security

Portions courtesy Ellen Liu
Outline

• Introduction
• How security is compromised
• Security tips
• Security power tools
• Potpourri
Introduction

• **Computer Security** - protection of an automated information system in order to preserve the integrity, availability and confidentiality of information system resources, including hardware, software, firmware, information/data, and telecommunications

• **CIA Triad**
  
  – **Confidentiality**: Data confidentiality, privacy
  
  – **Integrity**: Data integrity, system integrity, **authenticity**: origin integrity, **accountability/non repudiation**: ability to trace a security breach to a responsible party

  – **Availability**
General Consensus

• No OS is secure. Security breaches are commonplace

• Need patience, vigilance, knowledge, persistence from all user, admin, management communities

• Security is an ongoing battle that can never really be won

• Security can make system more resistant to attacks

• Security often means less convenience and more constraints to users
How Security is Compromised

There are many vulnerabilities, threats, risks, and attacks. We will focus on just three aspects:

- Social engineering
- Software vulnerability
- Configuration errors
Social Engineering

• Seemingly legitimate personnel or colleague ask for info
• **Phishing**: collect info via deceptive emails, instant msgs
• Often provide victim-specific info gleaned elsewhere to appear authentic and earn trust

• Need site policies on phone dos and don’ts, physical security, password selection, etc.
• Many organizations inform users that administrators will never request their passwords. Report immediately if such incidents occur
Software Vulnerabilities

- Program errors or context dependencies
- **Buffer overflow**: allocate a fixed-size buffer to store data, without checking the actual size of data to be stored. If larger than buffer size, it overflows /overwrites adjacent memory space, may crash the program or execute arbitrary code
  - Some programming systems include automatic checks
- **Input validation vulnerabilities**

```perl
#!/usr/bin/perl
open(htmlfile, "'/var/www/html/$argv[0]"") or die "fail\n";
while(<htmlfile>) { print; }
close htmlfile;

$argv[0] is a user input. What if sb enters .../.../.../etc/passwd
```
SQL Injection.

User-Id: srinivas
Password: mypassword

```
select * from Users where user_id= 'srinivas'
    and password = 'mympassword'
```

User-Id: `OR 1= 1; /*`
Password: `*/--`

```
select * from Users where user_id= `\' OR 1 = 1; /*`
    and password = `*/--`
```

Evaluation of 1=1 will always be true
/* */ enclose comments
-- precedes a comment within a single line
Configuration Errors

• Security vs. convenience
  – E.g., accounts without passwords, disks shared with the world, unprotected databases

• Boot loader password example
  – GRUB can be configured at install time to require a password, admins almost always decline the option
  – This leaves the system open to physical attack
  – With a password means if the system is rebooted, say, after a power outage, an admin has to drive to work to get the machine up and running again

• Do not leave ports open
CentOS Entering GRUB
Security Tips

- Patches
- Unnecessary services
- Remote event logging
- Backups
- Malware (viruses, worms, Trojans, rootkits)
- Packet filtering, passwords, vigilance
Patches

• Keeping the system updated with the latest patches is chore of the highest security value

• A recommended patching approach includes:
  – A regular schedule to install routine patches
  – A change plan to document impact, post-installation testing steps, and steps to back out the changes if needed
  – Understand what patches are relevant
    • Keep an inventory of apps and OS in use
    • Subscribe to vendor-specific lists/blogs, also general ones such as Bugtraq
Unnecessary Services

• Find out which services are running
  – Use the `netstat` command to find all listening sockets
• Find and identify services that use unknown ports
  – Use the `fuser`, `lsof`, and then `ps` commands
• If not needed, stop it, and do not start it at boot time
• Disable known vulnerable network protocols
  – FTP, Telnet
  – BSD “r” programs: rcp, rlogin, rsh
Disable root ssh login

• Sudo is good enough
• A high-value target for brute-force guessing
• In /etc/ssh/sshd_config:

  PermitRootLogin no
Remote Event Logging

- **Syslog** forwards log info to files, lists of users, or other hosts on network
- Set up a secure host as a **central logging** machine
  - Parse forwarded events and take proper action such as alerting admins when certain events occur
- Remote logging also prevents hackers from covering their tracks by rewriting or erasing log files on compromised systems
Backups

- Regular backups of all partitions and store some backups off-site

- When storing tapes off-site, use a fireproof safe to deter theft, also use encryption
  - If using contract storage facility, take a physical tour
Viruses and Worms

- **Viruses**: Rogue software program that attaches itself to other software programs or data files in order to be executed
- **Worms**: Independent programs that copy themselves from one computer to other computers over a network
- **Linux/UNIX** have been mostly immune from viruses
  - Less market share in desktop market, thus not a target
  - Access control in Unix may limit self-propagating worm or virus; need root privilege to alter system executables
Don’t neglect email and file servers!

• A Linux server can inadvertently distribute viruses to Windows machines on the network

• Run antivirus software on UNIX servers to protect site’s Windows systems from Windows viruses
  – E.g., mail server scans inboxes, file server scans shared files
  – Supplement with desktop antivirus such as ClamAV: a popular, free antivirus product with signatures of thousands of viruses
Trojan Horses

- **Trojan horses**: programs that aren’t what they seem to be. E.g., claims to draw a picture, but deletes files instead

- Packages affected in the past
  - sendmail, tcpdump, OpenSSH, InterBase
  - Typically embed code that allows attackers to access the victim’s systems at will
  - Fixed in a week or two, notified in mailing list

- Obvious security problems are discovered quickly and widely discussed on the net
  - Google a software package before installing it
Rootkits

• **Rootkits**: programs and patches that enable continued privileged access to a computer while hiding important system information such as process, disk, or network activity
  – Cover tracks and avoid detection
  – So the attacker can continue using the system to distribute software illegally, probe other networks, or launch attacks against other systems
  – Range from hacked `ls` and `ps`, to hacked kernel modules

• **Tools to detect**: host-based IDS e.g., **OSSEC**, special scripts e.g., **chkrootkit**

• Compromised machine is better reformatted than cleaned
Packet Filtering, Passwords, Vigilance

• **Packet filtering:** always filter network packets entering the system
  – Use packet-filtering routers, firewall, or filter software

• **Passwords**
  – every account must have a hard-to-guess password
  – Never send plaintext reusable passwords across the net
  – Always use secure remote access software such as ssh

• **Vigilance**
  – Monitor system health, network connections, process table, status report regularly (daily)
  – Perform regular self-assessment
Security Power Tools

**Warning:** Do not run these tools on someone else’s system or network without permission! Instead use them for self-assessment/debugging.

- Port Scanner: Nmap, Nessus
- Password Cracker: John the ripper
- Network IDS – Bro, Snort
- HIDS: OSSEC
Nmap

• A network port scanner
• Check a set of target hosts to see which TCP and UDP ports have servers listening on them
• A port is a numbered communication channel
  – An IP address identifies an entire machine
  – An IP address + a port # identifies a server, an application, or a conversation on that machine
  – Most network services are associated with “well known” port numbers. See /etc/services

IANA up-to-date port list: http://www.iana.org/assignments/port-numbers
Nmap Output

[root@darkstar ~]#
[root@darkstar ~]# nmap -PN -ss -O Scanme.Nmap.Org

Starting Nmap 5.21 ( http://nmap.org ) at 2010-04-01 11:19 IDT
Nmap scan report for Scanme.Nmap.Org (64.13.134.52)
Host is up (0.18s latency).
rDNS record for 64.13.134.52: scanme.nmap.org
Not shown: 993 filtered ports

<table>
<thead>
<tr>
<th>PORT</th>
<th>STATE</th>
<th>SERVICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>25/tcp</td>
<td>closed</td>
<td>smtp</td>
</tr>
<tr>
<td>53/tcp</td>
<td>open</td>
<td>domain</td>
</tr>
<tr>
<td>70/tcp</td>
<td>closed</td>
<td>gopher</td>
</tr>
<tr>
<td>80/tcp</td>
<td>open</td>
<td>http</td>
</tr>
<tr>
<td>113/tcp</td>
<td>closed</td>
<td>auth</td>
</tr>
<tr>
<td>8009/tcp</td>
<td>open</td>
<td>ajp13</td>
</tr>
<tr>
<td>31337/tcp</td>
<td>closed</td>
<td>Elite</td>
</tr>
</tbody>
</table>

Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.15 - 2.6.26

OS detection performed. Please report any incorrect results at http://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 16.99 seconds
[root@darkstar ~]#
Interpreting Nmap Output

- The host Scanme.Nmap.Org is running three services: 53, 80, and 8009. Under “STATE”
  - open: ports that have servers listening
  - closed: ports with no server
  - unfiltered: ports in an unknown state
  - filtered: cannot be probed due to intervening packet filters

- May guess what OS is used based on implementation of TCP/IP

- May guess what software is behind a running open port
Nessus

- **Nessus**: The most widely accepted and complete vulnerability scanner available
  - Scans for network servers running on any port and checks for known vulnerabilities instead of relying on version numbers

- Closed source, proprietary, but freely available

- New vulnerability checks (called **plugins**) daily, freely available to non-commercial users
John the Ripper

- A finder of insecure passwords from Solar Designer
- Implements several password-cracking algorithms
- It replaces an earlier tool called crack
- Can scan encrypted password files e.g., /etc/shadow

```
root@undecided:#echo john /etc/shadow
Loaded 3 password hashes with 3 different salts (FreeBSD MD5 [32/32])
badpass            (tjones)
test               (test)
```

- Again, do not try it against others’ passwords without approval
What makes a secure password?

• Hard to guess

• If I were an attacker, what would I guess first?
  
  – User name
  
  – Dictionary words
  
  – Oh, and I’d do obvious special character substitutions
    
    • 5 for an s, @ for an a, etc.

• What is the best password?
  
  – A truly random string

• How do I construct randomness?
Truth from xkcd

Through 20 years of effort, we've successfully trained everyone to use passwords that are hard for humans to remember, but easy for computers to guess.
True randomness

• Humans can’t produce random passwords. Let a program do it:
  – Diceware aggregates common words for important passwords
  – Lastpass generates un-rememberable passwords, has browser plugins

• Also, note that having published requirements, like “must have exactly one number” or “six to eight characters” can actually limit the search space of the attacker
  – If possible, best to keep private to your users
Requiring strong passwords

- We’ve discussed PAM previously
- It has a nice module pam-cracklib that can reject weak passwords
- Add to /etc/pam.d/common-password:
  
  ```
  password requisite pam_cracklib.so retry=3 minlen=8 difok=3
  ```
Aging passwords?

- “You must change your password every 3 months”
- Good idea?
  - Pros: Mitigate risk of a very slow brute-force attack
  - Cons: Users dislike having to come up with new passwords, more likely to reuse a password
Bro

• **Bro**: An open source network intrusion detection system (NIDS), monitors network traffic and looks for suspicious activities

• Inspects all traffic into and out of a network
  – **Passive mode**: report on suspicious activity
  – **Active mode**: injects traffic to disrupt malicious activity

• **Sophisticate**: correlate inbound and outbound traffic

• **Configuration** is complex and require good coding experience

• **Capable**: can supplement or replace a commercial NIDS
Snort

• An open source NIDS and network IPS (intrusion prevention system). Basis for many commercial NIDS implementations

• Free base, subscription fee to access the most recent detection rules
  – Third-party extensions. E.g., Aanval

• Signature (i.e., a set of rules extracted for known attacks) based

• Less powerful than Bro, but much simpler to configure
  – A good “starter” NIDS
OSSEC

• **Host-based intrusion detection (HIDS).** Free software
  – Rootkit detection
  – Filesystem integrity checks
  – Log file analysis
  – Time-based alerting and active responses

• **Monitors host activity, takes action according to a set of rules configured**

• **Two components**
  – **The manager** (server): one per network. It stores file-integrity check databases, logs, rules, configurations, events, auditing entries
  – **Agents** (clients): on each host and reports to the manager
Potpourri

- Setuid
- Chroot
- Mandatory Access Control and SELinux
- SSH tunneling
- What to do if you are attacked?
Setuid-to-Root Binary

$ stat -c 'Access:(%a/%A) Uid:(%u/%U) Gid:(%g/%G)' /bin/bash
Access: (0755/-rwxr-xr-x) Uid: (0/root) Gid: (0/root)

$ stat -c 'Access:(%a/%A) Uid:(%u/%U) Gid:(%g/%G)' /bin/mount
Access: (4755/-rwsr-xr-x) Uid: (0/root) Gid: (0/root)

• Setuid causes a binary to run as the file owner, rather than the user that issued the command

• Trusted setuid binaries export safe functionalities.
  – E.g., ping, mount, passwd, etc.
  – Administrator configures policies on safe subsets
/* Parse arguments */
if (ruid == 0 ||
    user_mount_ok(args))
    sys_mount(args);

// Kernel
......

/* Setuid to Root */
User
/etc/fstab

/dev/cdrom /mnt/cdrom
iso9660 user,ro 0 0

/sys/mount /dev/cdrom /mnt/cdrom
/* Exploit vulnerability */

fd = sys_open(/etc/shadow);

sys_mount() {
    if(!capable(CAP_SYS_ADMIN))
        return -EPERM;
    Kernel ...... 
}

sys_open() {
    if(!perm_check(file, euid))
        return -EPERM;
    Kernel ...... 
}

/* Parse arguments */

if(ruid == 0 || user_mount_ok(args))
    sys_mount(args);

User
/etc/fstab
/etc/shadow

/bin/mount /dev/cdrom /mnt/cdrom

Setuid to Root

Vulnerable mount
Principle of Least Privilege

- Least authority necessary to perform duties
- Setuid-root violates least privilege principle
  - Empowers binaries to issue privileged system calls
- Kernel policy conflicts with the system policy
  - Kernel: only root can mount
  - System: any user can mount at safe locations
- Setuid binary mount bridges the gap
Advice

• Think twice before installing setuid-root programs
  – Some are required, but I would minimize this

• Mount non-root file systems with nosuid
  – Avoid someone adding a setuid binary from a cdrom or flash drive
chroot

• Confine a process to a given directory
• Useful for sandboxing (or jailing) a program
  – Although you do have to create a complete environment
• Other useful tools to sandbox an application:
  – Chromium sandbox, plash, etc.
Mandatory Access Control

• Mandatory Access Control (MAC)
  – have the control of all permissions in the hands of a security administrator
  – Do not allow users to modify any permissions, even on their own objects. Contrast traditional Unix access control

• Users are assigned a security level from a structured hierarchy. Users can read/write items at the same level or lower, but not any higher level
  – User with “secret” access cannot read “top secret” objects

• Least privilege - allowing access only when necessary
  – Limit scope of breach to specific resources required by SW
SELinux

• MAC is available to UNIX and Linux
  – Solaris trusted extension, HP-UX security containment, etc.
• Security-enhanced Linux (SELinux)
  – Implements MAC for Linux. Default component in Red Hat 4+
  – Adopted in environment with strict security requirements. E.g., government agencies
  – Policy is critical. E.g., to protect a daemon, a policy must enumerate all files, directories, and other objects to which the process needs access.
  – /etc/selinux/config controls SELinux configuration. Check /var/log/messages for SELinux errors, if problems with newly installed software
SELinux Administration

• SELinux is used by Fedora,
  – Users tolerate it mostly because they have good defaults
• Make no mistake: writing SELinux policies is **hard**
  – If you have a one-off piece of software, you will probably pay RedHat consultants to write a policy for you
• Still not a bad idea...
SSH tunneling

• A common firewall setting: Only let ssh in
• What if I want to access a web server behind a firewall?
• SSH to the rescue!
An SSH Tunnel

Type localhost:2000 as url
Why SSH tunnels are ok

• Still only expose ssh to outside world
• An authorized user can connect to services inside a firewall from a computer inside the firewall
• No risk beyond allowing ssh in the first place
• Fairly easy to configure (previous example):

```bash
ssh -f user@example.com
   -L 2000:internal-webserver.example.com:80 -N
```
Final advice

• Subscribe to mailing lists for software you administer
• They announce important security patches you may want to push out more aggressively
  – E.g., “This specially crafted packet to ssh drops you to a root shell”
What to do when your site is attacked

9-step plan

• Don’t panic
• Decide on an appropriate level of response
• Collect away all available tracking information
• Assess degree of exposure
• Pull the plug
• Devise a recovery plan
• Communicate the recovery plan
• Implement the recovery plan
• Report the incident to authorities