Hackers / Intruders

• External attacks
  – Typical “hacker”
  – Exploits carried out remotely
  – Does not have an account on the remote machine

• Insider attacks
  – Carried out from within
  – Attacker may have an account on the attacked system
  – E.g. Disgruntled employees, spies, knowledgeable former employees

• Both can make effective use of:
  – Social engineering
  – System vulnerabilities
External Attacks

• Attacks usually consist of several stages:
  – Finding software vulnerabilities
  – Exploiting them
  – Hiding/cleaning up the exploit

• Finding vulnerabilities:
  – What machines are available?
  – What OS / version / patch level are the machines running?
  – What additional software is running?
  – What is the network topology?
  – How detectible will the attack be?
  – How can the attacker cover her tracks?
Attacker Reconnaissance

• **Network Scanning**
  – Existence of machines at IP addresses
  – Attempt to determine network topology
  – ping, tracert

• **Port scanners**
  – Try to detect what processes are running on which ports, which ports are open to connections.
  – Typical machine on the internet gets 10-20 port scans per day!

• **Web services**
  – Use a browser to search for CGI scripts, Javascript, etc.
Defenses

• Disable ping service, unneeded ports
  – Not always possible (you do want some connectivity!)

• Firewalls and NATs
  – Filter out inappropriate packets
  – Scanners use broadcast, multicast packets, clever port/flag combinations to thwart firewall filters

• Keep audit logs of requests
  – Generates a lot of data, hard to sift through
  – Clever port scan packages use a “drip scan” approach, sprinkling their scan packets sparsely across several hours or days
Determining OS information

- Gives a lot of information that can help an attacker carry out exploits
  - Exact version of OS code can be correlated with vulnerability databases
- Sadly, often simple to obtain this information:
  - Just try telnet

```
playground~> telnet hpux.u-aizu.ac.jp
Trying 163.143.103.12 ...
Connected to hpux.u-aizu.ac.jp.
Escape character is '^]'.
HP-UX hpux B.10.01 A 9000/715 (ttyp2)
login:
```
Determining OS

• Or ftp:

```
$ ftp ftp.netscape.com 21
Connected to ftp.gftp.netscape.com.
220-36
220 ftpnscp.newaol.com FTP server (SunOS 5.8) ready.
Name (ftp.netscape.com:stevez):
331 Password required for stevez.
Password:
530 Login incorrect.
ftp: Login failed.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> system
215 UNIX Type: L8 Version: SUNOS
ftp>
```
Determining OS

• Exploit different implementations of protocols
  – Different OS’s have different behavior in some cases

• Consider TCP protocol, there are many flags and options, and some unspecified behavior
  – Reply to bogus FIN request for TCP port
    (should not reply, but some OS’s do)
  – Handling of invalid flags in TCP packets
    (some OS’s keep the invalid flags set in reply)
  – Initial values for RWS, pattern in random sequence numbers, etc.
  – Can narrow down the possible OS based on the combination of implementation features

• Tools can automate this process
Nmap screen shot

http://www.insecure.org/nmap
http://www.insecure.org/nmap/nmap-fingerprinting-article.html
Exploiting Vulnerabilities

• Format string vulnerabilities
  – Command injection
  – SQL injection against databases

• Application-level DoS
  – Make use of application knowledge

• Network-level DoS

• Look for buffer overflows
  – OS/architecture information is useful to craft appropriate attack data
Covering tracks

- Use network topology information to exploit nearby machines
- Hiding attacker’s identity
  - Use uplink to unprotected wireless (802.11) networks
  - Tunnel through multiple administrative domains in different regions (e.g. go Starbucks -> China -> Russia -> U.S.)
  - Distributed ISP’s can’t easily share audit logs.
- If exploit is successful
  - use secure software!
  - SSH instead of Telnet (means attacker actions can’t be observed)
  - Use root privileges to clean up logs
  - Reboot machines that may have crashed
  - Install rootkit / modify kernel so that further audits won’t detect it
Rootkits

• Patches or Kernel code that provides attackers a back door

• Override some kernel behavior
  – Replace WinNT’s QueryDirectoryFile() kernel call with one that hides files and directories that begin with a particular string (e.g. “_root_”)
    • Installed as a Device driver
  – Binary patch to kernel code for SeAccessCheck()
    • Critical function for the reference monitor in WinNT
    • A 2 bit change can disable all access controls!

• Countermeasures?
  – Secure hardware / secure coprocessors
  – Not very common
Reflections on Trusting Trust

• If the TCB is compromised, then it is almost impossible to detect it.
• Ken Thompson article:
  – http://www.acm.org/classics/sep95/

• Modify a compiler so that it incorrectly compiles the “login” program, but otherwise does the right thing
• Compile the compiler with itself (“bootstrap”)
• Completely hides the attack inside the executable—extremely hard to detect
Insider Attacks

• Trojan Horses
  – Planted by successful external attack
  – Purposefully added by internal attacker

• Out-of-band attacks
  – Simply print the secret document and put it in your briefcase
  – Swap the hardware
Salami Attacks

• Programs that compute money are targets of this attack
  – Use subtle rounding errors
  – Often distribute errors among many transactions
  – Accumulate small amounts of money

• Example:
  – 6.5% (annual) interest on $102.87 for 31 days is $0.5495726
  – Instead of rounding to $0.55, round to $0.54
  – Funnel the extra penny somewhere else
Covert Channels

• Program that leaks confidential information intentionally via secret channels.

• Not that hard to leak a small amount of data
  – A 64 bit shared key is quite small!

• Example channels
  – Adjust the formatting of output: use the “\t” character for “1” and 8 spaces for “0”
  – Vary timing behavior based on key
Differential Power Analysis

- Read the value of a DES password off of a smartcard by watching power consumption!

- This figure shows simple power analysis of DES encryption. The 16 rounds are clearly visible.
TEMPEST Security

• Transient Electromagnetic Pulse Emanation Standard
  – (Or?) Temporary Emanation and Spurious Transmission
  – Emission security (Van Eck phreaking)
  – computer monitors and other devices give off electromagnetic radiation
  – With the right antenna and receiver, these emanations can be intercepted from a remote location, and then be redisplayed (in the case of a monitor screen) or recorded and replayed (such as with a printer or keyboard).
TEMPEST

• Policy is set in National Communications Security Committee Directive 4

• Guidelines for preventing EM reception
  – Shield the device (expensive)
  – Shield a location (inconvenient?)

• Not a risk?
  – Most of the guidelines are classified!
Defenses for Covert Channels

• Well specified security policies at the human level
• Auditing mechanisms at the human level
  – Justify prosecution if the attacker is caught
• Code review
  – This is a form of audit
• Automated program analysis
  – Type systems that let programmers specify confidentiality labels

• But, not much you can do against a determined insider