Announcements

• Midterm I will be held Friday, Oct. 6th.

• Project 2 is on the web

• Project 1 has been graded:
  – Class average: 82
  – Std. Dev. 8

• Solutions for HW 1 are on the web
Principles for Firewall Configuration

• Least Privileges:
  – Turn off everything that is unnecessary (e.g. Web Servers should disable SMTP port 25)

• Failsafe Defaults:
  – By default should reject
  – (Note that this could cause usability problems…)

• Egress Filtering:
  – Filter outgoing packets too!
  – You know the valid IP addresses for machines internal to the network, so drop those that aren’t valid.
  – This can help prevent DoS attacks in the Internet.
Another problem with Filtering

• Handling IP Fragments
  – Possible for ACK and SYN flag bits in a TCP packet could end up in a different IP fragment than the port number
  – There are malicious tools that intentionally break up traffic in this way
  – Fix: Problem is "tiny" initial IP fragment, so require that initial IP fragment be > 16 bytes (or better yet, large enough for whole TCP header).
Proxy-based Firewalls

- Proxy acts like both a client and a server.
- Able to filter using application-level info
  - For example, permit some URLs to be visible outside and prevent others from being visible.
- Proxies can provide other services too
  - Caching, load balancing, etc.
  - FTP and Telnet proxies are common too
Benefits of Firewalls

• Increased security for internal hosts.
• Reduced amount of effort required to counter break ins.
• Possible added convenience of operation within firewall (with some risk).
• Reduced legal and other costs associated with hacker activities.
Drawbacks of Firewalls

• Costs:
  – Hardware purchase and maintenance
  – Software development or purchase, and update costs
  – Administrative setup and training, and ongoing administrative costs and trouble-shooting
  – Lost business or inconvenience from broken gateway
  – Loss of some services that an open connection would supply.

• False sense of security
  – Firewalls don’t protect against viruses…
Network Vulnerabilities

• Anonymity
  – Attacker is remote, origin can be disguised
  – Authentication

• Many points of attack
  – Attacker only needs to find weakest link
  – Attacker can mount attacks from many machines

• Sharing
  – Many, many users sharing resources

• Complexity
  – Distributed systems are large and heterogeneous

• Unknown perimeter

• Unknown attack paths
Syn Flood Attack

• Recall TCP’s 3-way handshake:
  – SYN    ---  SYN+ACK  --- ACK

• Receiver must maintain a queue of partially open TCP connections
  – Called SYN_RECV connections
    – Finite resource (often small: e.g. 20 entries)
    – Timeouts for queue entries are about 1 minute.

• Attacker
  – Floods a machine with SYN requests
  – Never ACKs them
  – Spoofs the sending address (Why? Two reasons!)
Reflected denial of service

• Broadcast a ping request
  – For sender’s address put target’s address
  – All hosts reply to ping, flooding the target with responses

• Hard to trace
• Hard to prevent
  – Turn of ping? (Makes legitimate use)
  – Limit with network configuration by limiting scope of broadcast messages
(Distributed) Denial of Service

• Coordinate multiple subverted machines to attack
• Flood a server with bogus requests
  – TCP SYN packet flood
  – Up to 600,000 packets per second
• Detection & Assessment?
  – 12,800 attacks at 5000 hosts in 3 week period!
  – IP Spoofing (forged source IP address)
• Prevention?
  – Filtering?
  – Decentralized file storage?
Malicious Code

- Trapdoors (e.g. debugging modes)
- Trojan Horses (e.g. Phishing, Web sites with exploits)
- Worms (e.g. Slammer, Sasser, Code Red)
- Viruses (e.g. Bagle MyDoom mail virus)

- The distinction between worms and viruses is somewhat fuzzy
Trapdoors

• A trapdoor is a secret entry point into a module
  – Affects a particular system

• Inserted during code development
  – Accidentally (forget to remove debugging code)
  – Intentionally (maintenance)
  – Maliciously (an insider creates a hole)
Trojan Horse

- A program that pretends to be do one thing when it does another
  - Or does more than advertised

- Login Prompts
  - Trusted path

- Accounting software

- Examples:
  - Game that doubles as a sshd process.
  - Phishing attacks (Spoofed e-mails/web sites)
Worms (In General)

• Self-contained running programs
  – Unlike viruses (although this distinction is mostly academic)

• Infection strategy more active
  – Exploit buffer overflows
  – Exploit bad password choice

• Defenses:
  – Filtering firewalls
  – Monitor system resources
  – Proper access control
Viruses

• A *computer virus* is a (malicious) program
  – Creates (possibly modified) copies of itself
  – Attaches to a host program or data
  – Often has other effects (deleting files, “jokes”, messages)

• Viruses cannot propagate without a “host”
  – Typically require some user action to activate
Virus/Worm Writer’s Goals

- Hard to detect
- Hard to destroy or deactivate
- Spreads infection widely/quickly
- Can reinfect a host
- Easy to create
- Machine/OS independent
Effects of Malicious Code

• Data corruption
• Denial of service
  – Crash machines
  – Overload network infrastructure
• Expose confidential / secret information
• Create "zombie" devices
  – Allows attacker to use other people's machines, often for spam, or distributed denial of service