# Security: Attack and Defense

Aaron Hertz
Carnegie Mellon University

15-441 Computer Networks
Spring 2003

## Outline

- Breaking into hosts
- DOS Attacks
- Firewalls and other tools

## Breaking Into Hosts

- Guessing Passwords
- Port scans
- Stack overflow
- TCP Hijacking

## Identify Targets

- Is a host alive?
  - Use ping (ICMP ECHO request and reply)
- Is a host running, say, a telnet server?
  - Port scan (most servers listen on well-known ports)
    - TCP: try connect()on all ports (ECONNREFUSED)
    - UDP: try sendto() on all ports (ICMP_UNREACH_PORT)
  - “Stealth Scan”
    - E.g. nmap (www.insecure.org)
- What OS is a host running?
  - Different OSes react differently to special packets

# Popular Port Scanners

- NMAP – http://www.insecure.org/nmap
  - TCP scans (full 3-way handshake on every port)
  - UDP scans
  - SYN scans using IP fragments
  - ACK and FIN scans
  - Designed to by-pass firewalls and intrusion detection systems
- QueSO – http://www.apostols.org/projectz/queso
  - TCP scans with various combinations of TCP flags: SYN, SYN+ACK, FIN, FIN+ACK, SYN+FIN
  - Can determine remote hosts operating system, even kernel version

# Gain Access

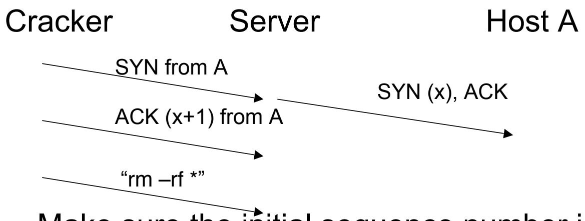
- Direct Access
  - Backdoor
  - Use passwords obtained from packet sniffing
  - Password guessing
    - E.g. use a dictionary attack on /etc/passwd
  - Bribery, blackmail, torture, etc.
- Exploit vulnerability to gain access
  - Protocol vulnerability
    - E.g. TCP sequence number prediction
  - Software vulnerability
    - E.g. buffer overflow, format string, etc.

# Backdoors

- Secret way into the system, bypassing normal authentication
- Usually left by original programmers, though could be a result of someone compromising the code base
- Having the source doesn't guarantee immunity
  - See "Reflections on Trusting Trust"

# TCP Sequence Number Prediction

- Problem if a server uses IP/hostname based authentication
  - E.g. ".rhost" for rlogin



Cracker Server Host A

SYN from A

SYN (x), ACK

ACK (x+1) from A

"rm –rf *"

  - Make sure the initial sequence number is "hard" to predict

# Session Hijacking

- Allows an attacker to steal, share, terminate, monitor and log any terminal session that is in progress
- Session stolen across the network
- What can be hijacked:
    - telnet, rlogin, rsh, ftp
  - Simple session hijacking scenario:
    - A telnets to B to get some work done
    - Attacker resets connection to A
    - Attacker kicks off A and takes over the session to B

# Buffer Overflows

- One of the most used “hacking” techniques
- Advantages
  - Very effective
    - Attack code runs with privileges of exploited process
  - Can be exploited locally and remotely
    - Interesting for network services
- Disadvantages
  - Architecture/OS version dependent
    - Directly inject assembler code / call system functions
  - Some guess work involved (correct addresses)

# Stack Overflow Attack

- Data is copied into local variables without proper bound checking
  - Vulnerable functions: strcpy, strcat, gets, fgets, sprintf…
- Data “overflows” allocated buffer and overwrites stack data (especially return address)
  - If done with random data, usually causes a segmentation fault
- Carefully overwrite content and set return value to user-defined value
  - Causes a jump to user-defined code – modified execution flow
  - This code is executed with privileges of running process

# Stack Overflow: Code

- What code should be placed in the buffer?
  - Assembly instructions, system calls, alignment
  - Different variations for different platforms
  - Do not know addresses
- Usually, a shell in started
  - Use system call (execve) to spawn shell
  - Runs with same privileges as exploited application

# Social Engineering

- An attempt by a hacker to persuade a legitimate system user to reveal information
- Most common way hackers break into systems
- “If you give me your logon ID and password, I can fix it in a few minutes, you can change your password when I am done”….
  - A real help desk employee will never ask for this!
- Hacker takes advantage of the organization size – people do not know each other
- Ignorance is a big help to the attacker

# After Gaining Access

- Obtain confidential information
  - E.g. emails, credit card numbers, et.
- Destroy files, prevent login, …
- Use the host as a base for future attacks
  - Use it for a DDoS attack
  - Use it to gain access to other machines in a corporate network
  - Install “rootkit”: modified system tools, for example:
    - ps: won’t display certain processes
    - ls: won’t display certain files
    - netstat: won’t display certain network connections
  - Run packet sniffer to obtain more information (e.g. passwords)
  - …

# Detecting Attacks: Intrusion Detection

- What to detect?
  - Intrusion attempts
  - Successful intrusions – compromised hosts
- Detecting intrusion attempts
  - Filter and log certain packets
  - Analyze the logs
  - Example: snort
    - http://www.snort.org

# Bypassing Intrusion Detection Systems

- Sneak attacks past an IDS
  - Fragmentation
  - HTTP non-standard URL encodings
    - ‘/’ padding: /cgi-bin///phf
    - Self-referencing directories: /cgi-bin/./phf
    - URL encodning: %2fcgi-bin/phf
    - Reverse directory traversal: /cgi-bin/here/../phf
- False alarms
  - Fill security logs with many false attacks, so real attacks go un-noticed

## Honeypots

- Fake machine designed to appear interesting to hackers
  - Emulated services that appear vulnerable
  - May contain fake confidential data
- Any interaction with the honeypot is unauthorized
- Uses
  - Track hackers
  - Research new attack methods

## Network Telescopes

- Chunk of globally-routable IP addresses with little or no legitimate traffic
- Used to “see” remote security events
  - Attacks directed at random target addresses
    - Especially worms, e.g. Code Red
  - Backscatter from DoS attacks
    - Attacker must **randomly** spoof source address
      - True of most major attack tools
      - Not SMURF or other reflector attacks
    - Received backscatter is evidence of an attack elsewhere

## Detecting Compromised Hosts

- Check for the presence of a “rootkit”
- “Integrity check”
  - Construct a database that stores a signed hash of each important file
  - Check all files periodically (e.g. every day)
  - Example: tripwire
    - http://www.tripwire.org

## Denial of Service Attacks

- Make services unavailable
- Typically achieved by wasting resources associated with the service
  - Network bandwidth, memory, CPU cycles
  - Challenge: make the defense cheap
- Common attacks:
  - SYN attack, SMURF, DDoS
- IP traceback

# Examples of DoS Attacks

- There are countless DoS attacks out there today
  - http://www.cert.org/tech_tips/denial_of_service.html
- Various forms:
  - SYN Flooding
  - Land (and similar)
  - Teardrop (and similar)
  - Smurf, Papasmurf
  - Ping of Death

# DoS: TCP SYN Flooding

- TCP is subject to SYN Flooding
- TCP based on 3-way handshake (ISN – initial sequence number)
  - A -----SYN(A, ISN$_A$)-----------------------> B
  - A <---ACK(A, ISN$_A$), SYN(B, ISN$_B$)----- B
  - A -----ACK(B, ISN$_B$)-----------------------> B
- System must allocate resources for each SYN which arrives
- SYN attack scenario
  - Attacker sends several SYN packets to a victim from a spoofed (fake) machine SYN(X, ISN$_X$)
  - Connection is never ACK'd, and waits for timeout
  - Victim's queue fills up, either crashes or cannot serve more requests

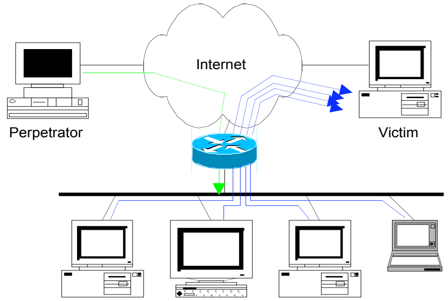
# Defense against SYN Flooding

- SYN Cookies: Clever way to accept new connections when SYN queue is full
- Built into Linux and FreeBSD
- ISN is a function of several variables:
  - Top 5 bits: time stamp
  - Next 3 bits: encoding of MSS based on client's MSS
  - Bottom 24 bits: cryptographically secure function of client and server addresses and port numbers and timestamp
- Can rebuild a dropped SYN from information encoded in ACKed sequence number
- But, cryptography means connections cannot be forged

# SMURF

🟩 ICMP echo (spoofed source address of victim)
Sent to IP broadcast address

🟦 ICMP echo reply



# SMURF Defense

- Not much, at target
  - Even if you block the packets, upstream bandwidth still clogged
- To prevent a SMURF from originating from a network:
  - Exit filtering for spoofed packets
  - Disallow incoming ICMP packets to broadcast addresses

# Firewalls

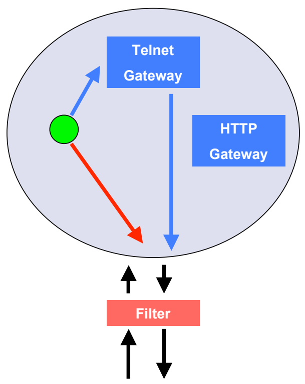
- The goal of the firewall is to control what traffic enters and leaves a network
  - Creates a trust boundary: people outside of the firewall are trusted less than people inside the firewall
  - Similar to putting a guard at the door checking IDs
- Firewalls alone do not offer sufficient security
  - Still have to be concerned about security breaches within the organization
  - Every organization has materials that require different levels of secrecy
  - But, firewalls limit how much traffic has to be monitored
  - Can also help with DoS attacks

# Filter-based Gateways

- A filter classifies packets based on the header
  - IP addresses
  - Port numbers
  - Protocol and message types
  - Connection information
- Filter decides which packets go through and which packets are dropped.
  - No telnet, only outgoing web connections, …

# Application Gateways

- The application-level connection is terminated at the gateway and a separate connection is established over the external network
- The gateway can monitor contents of messages since it “understands” the application
  - Application header versus data
- Can be combined with the use of filters
  - E.g. the filter only forwards connections from an application gateway



# AAA

- Authentication, Authorization, Accounting
  - Process used whenever users access a commercial ISP
  - ISP wants to know who you are
  - ISP will verify that you are allowed to get service
  - ISP will want to keep track of your use of the network for charging and auditing purposes
- Example protocol is RADIUS
  - Example uses: dialup access to large access providers
  - IEFT standard