CS 526: Information Security

- Network Attacks
Network Protocols Stack

Application protocol

TCP protocol

Application

Transport

Network

Data Link

Network Access

IP protocol

Data Link

IP protocol

Network

Link

Data Link

Link
## Protocols

<table>
<thead>
<tr>
<th>Application</th>
<th>DNS, TFTP, TLS/SSL, FTP, Gopher, HTTP, IMAP, IRC, NNTP, POP3, SIP, SMTP, SNMP, SSH, TELNET, ECHO, BitTorrent, RTP, PNRP, rlogin, ENRP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BGP</td>
</tr>
<tr>
<td>Transport</td>
<td>TCP, UDP, DCCP, SCTP, IL, RUDP</td>
</tr>
<tr>
<td>Internet</td>
<td>OSPF, ICMP and IGMP</td>
</tr>
<tr>
<td></td>
<td>IP (IPv4, IPv6)</td>
</tr>
<tr>
<td></td>
<td>ARP and RARP</td>
</tr>
<tr>
<td>Network access</td>
<td>Ethernet, Wi-Fi, token ring, PPP, SLIP, FDDI, ATM, Frame Relay, SMDS</td>
</tr>
</tbody>
</table>
Types of Addresses in Internet

• MAC addresses in the network access layer
  – 48 bits or 64 bits
• IP addresses for the network layer
  – 32 bits for IPv4, and 128 bits for IPv8
  – E.g., 128.3.23.3
• IP addresses + ports for the transport layer
  – E.g., 128.3.23.3:80
• Domain names for the application/human layer
  – E.g., www.purdue.edu
Routing and Translation of Addresses

• Translation between IP addresses and MAC addresses
  – Address Resolution Protocol (ARP) for IPv4
  – Neighbor Discovery Protocol (NDP) for IPv6

• Routing with IP addresses
  – TCP, UDP, IP for routing packets, connections
  – Border Gateway Protocol for routing table updates

• Translation between IP addresses and domain names
  – Domain Name System (DNS)
Threats in Networking

• Confidentiality
  – Packet sniffing

• Integrity
  – Session hijacking

• Availability
  – Denial of service attacks

• Common
  – Address translation poisoning attacks
  – Routing attacks
Concrete Security Problems

- ARP is not authenticated
  - ARP spoofing (or ARP poisoning)
- Network packets pass by untrusted hosts
  - Packet sniffing
- TCP state can be easy to guess
  - TCP spoofing attack
- DNS is not authenticated
  - DNS poisoning attacks
Address Resolution Protocol (ARP)

- Primarily used to translate IP addresses to Ethernet MAC addresses
- Also used for IP over other LAN technologies, e.g., FDDI, or IEEE 802.11
- Each host maintains a table of IP to MAC addresses
- Message types:
  - ARP request
  - ARP reply
  - ARP announcement
ARP Spoofing (ARP Poisoning)

- Send fake or 'spoofed', ARP messages to an Ethernet LAN.
  - To have other machines associate IP addresses with the attacker’s MAC

- Defenses
  - static ARP table
  - detection: Arpwatch, DHCP snooping

- Legitimate use
  - redirect a user to a registration page before allow usage of the network
**Internet Protocol**

- **Connectionless**
  - Unreliable
  - Best effort

- **Transfer datagram**
  - Header
  - Data

---

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>4</td>
</tr>
<tr>
<td>Header Length</td>
<td>20</td>
</tr>
<tr>
<td>Type of Service</td>
<td>0</td>
</tr>
<tr>
<td>Total Length</td>
<td>6000</td>
</tr>
<tr>
<td>Identification</td>
<td>1234</td>
</tr>
<tr>
<td>Flags</td>
<td>0000</td>
</tr>
<tr>
<td>Fragment Offset</td>
<td>0</td>
</tr>
<tr>
<td>Time to Live</td>
<td>64</td>
</tr>
<tr>
<td>Protocol</td>
<td>TCP</td>
</tr>
<tr>
<td>Header Checksum</td>
<td>0x9135</td>
</tr>
<tr>
<td>Source Address</td>
<td>192.168.1.1</td>
</tr>
<tr>
<td>Destination Address</td>
<td>192.168.2.1</td>
</tr>
<tr>
<td>Options</td>
<td></td>
</tr>
<tr>
<td>Padding</td>
<td></td>
</tr>
<tr>
<td>IP Data</td>
<td>12345678901234567890123456</td>
</tr>
</tbody>
</table>

IP Routing

- Internet routing uses numeric IP address
- Typical route uses several hops
IP Protocol Functions (Summary)

- **Routing**
  - IP host knows location of router (gateway)
  - IP gateway must know routes to other networks

- **Fragmentation and reassembly**
  - If max-packet-size less than the user-data-size

- **Error reporting**
  - ICMP packet to source if packet is dropped
Packet Sniffing

- Promiscuous Network Interface Card reads all packets
  - Read all unencrypted data (e.g., “ngrep”)
  - ftp, telnet send passwords in clear!

Prevention: Encryption (IPSEC, TLS)
Tools for Network Sniffing

- tcpdump
- Windump
- Snort (network sniffer and network intrusion detection system)
- Wireshark (formerly Ethereal)
  - history of lot of buffer overflow vulnerabilities
- Sniffiy
- Dsniff
User Datagram Protocol

- IP provides routing
  - IP address gets datagram to a specific machine
- UDP separates traffic by port (16-bit number)
  - Destination port number gets UDP datagram to particular application process, e.g., 128.3.23.3:53
  - Source port number provides return address
- Minimal guarantees
  - No acknowledgment
  - No flow control
  - No message continuation
Transmission Control Protocol

- Connection-oriented, preserves order
  - Sender
    - Break data into packets
    - Attach packet numbers
  - Receiver
    - Acknowledge receipt; lost packets are resent
    - Reassemble packets in correct order
TCP Handshake

C

SYN_C

SYN_S, ACK_C

ACK_S

S

Listening

Store data

Wait

Connected
TCP Sequence Numbers

- Need high degree of unpredictability
  - If attacker knows initial seq # and amount of traffic sent, can estimate likely current values
  - Send a flood of packets with likely seq numbers
  - Attacker can inject packets into existing connection
- Some implementations are vulnerable
TCP Session Hijacking

- Each TCP connection has an associated state
  - Client IP and port number; same for server
  - Sequence numbers for client, server flows

- Problem
  - Easy to guess state
    - Port numbers are standard
    - Sequence numbers often chosen in predictable way
Risks from Session Hijacking

- Inject data into an unencrypted server-to-server traffic, such as an e-mail exchange, DNS zone transfers, etc.
- Inject data into an unencrypted client-to-server traffic, such as ftp file downloads, http responses.
- IP addresses often used for preliminary checks on firewalls or at the service level.
- Hide origin of malicious attacks.
- Carry out MITM attacks on weak cryptographic protocols.
  - often result in warnings to users that get ignored
- Denial of service attacks, such as resetting the connection.
Blind TCP Session Hijacking

- A, B trusted connection
  - Send packets with predictable seq numbers
- E impersonates B to A
  - Opens connection to A to get initial seq number
  - DoS B’s queue
  - Sends packets to A that resemble B’s transmission
  - E cannot receive, but may execute commands on A

Attack can be blocked if E is outside firewall.
Scanning

- Port scanning: The process of determining the services available on a computer, by sending packets to several ports
  - Useful to system administrators
  - Can also be used by attackers

- Vulnerability scanning: Communication on the ports having available services, for the purpose of determining vulnerabilities to available exploits
Scanning

- For an IP address, attacker aims to determine
  - which ports respond to queries
  - which vulnerabilities are present
- Often used before an attack is launched
  - Attack follows scanning about 50% of the time
- Scan probes a range of ports and IP addresses
  - *Footprint* of a scan = set of IP/port combinations
  - Independent of their order
Scan Classification

- **Horizontal** scan: 1 port, range of IP address
  - Attacker has 1 exploit, is interested in all hosts with the service corresponding to that exploit
- **Vertical** scan: 1 host, range of ports
  - Attacker is interested in a specific target
- **Block** scan = combines horizontal & vertical
- **Distributed** scan aims to evade blacklisting
  - Requests come from different IP addresses
Half-Open Scan (most common)

- Scanner sends a SYN packet
- A SYN-ACK response indicates an open port
  - Else a RST-ACK response
- Scanner responds to SYN-ACK with RST
  - Aborts the connection (incomplete handshake)
  - Without RST, it would timeout (and look DoS-like)
  - Leaves no trace in application-level logs (unlike using TCP connect(), which would be logged)
Port Scan Detection

• Distinguish between benign and scan traffic
  – Do so by using differences in their traffic patterns
• Benign TCP connection has SYN/FIN symmetry
  – TCP setup (3-step handshake: SYN, SYN-ACK, ACK)
  – Traffic after session is established
  – TCP tear-down (FIN)
• No such SYN/FIN symmetry in TCP scan traffic
  – Handshake is aborted (SYN, SYN-ACK, RST)
Scan Detection with Counting Bloom Filter

- k hash functions $H_1, \ldots, H_k$ (outputs $\leq n-1$)
- k arrays $C_1, \ldots, C_k$ of size n each, initialized to 0
- Upon seeing a SYN, increment by 1 every $C_i [H_i(\text{IP})]$ for $i = 1, \ldots, k$
- Upon seeing a FIN, decrement by 1 every $C_i [H_i(\text{IP})]$ for $i = 1, \ldots, k$
- For benign connection, all k counters are small
  - If large, the IP address is involved in a scan
Properties of Counting Bloom Filter Structure

• Can test whether a given element is stored
  – If all k counter values for it are nonzero then “Yes” (but not surely – false positives are possible)
  – No false negatives (a zero counter value implies a definite “No”)

• Well suited for implementing scan detection
  – Routers have limited memory and spare cycles

• Cannot list stored elements (but: not needed)
Is it Legal to Scan

• … if scan is not followed by an actual attack?
• Issue is not yet settled (“doorknob rattling”)
• Depends on ability to prove intent to attack
  – Intent is often erroneously believed to be present (e.g.,
    when legitimate scan accidentally hit an unintended
    web server and triggered an arrest)
  – Even when present, intent can be hard to prove
• Civil lawsuits easier to win (scanner pays $s)
SYN Flooding

C

SYN_{C1}

SYN_{C2}

SYN_{C3}

SYN_{C4}

SYN_{C5}

S

Listening

Store data
SYN Flooding

- Attacker sends many connection requests
  - Spoofed source addresses
- Victim allocates resources for each request
  - Connection requests exist until timeout
  - Old implementations have a small and fixed bound on half-open connections
- Resources exhausted $\Rightarrow$ requests rejected

- No more effective than other channel capacity-based attack today
Smurf DoS Attack

- Send ping request to broadcast addr (ICMP Echo Req)
- Lots of responses:
  - Every host on target network generates a ping reply (ICMP Echo Reply) to victim
  - Ping reply stream can overload victim

Prevention: reject external packets to broadcast address
SYN Flood Mitigation

• System configuration
  – Reduce the timeout to (e.g.) 10 seconds (drawback: denies access to legitimate but slower connections that require a higher timeout)
  – Increase the size of the queue (drawback: higher resource usage; can be flooded anyway)
  – Disable non-essential services, reducing the number of ports that can be attacked
SYN Flood Mitigation (cont’d)

- Router configuration, e.g.,
  - Block incoming packets that have source addresses from the internal network (an instance of ingress filtering)
  - Block outgoing packets that have source addresses from outside the internal network (an instance of egress filtering)
  - Issues with Mobile IP (mobile device has same IP address while it moves from one network to another)
Syn Flood Mitigation (cont’d)

- Monitor the TCP traffic within a local area network and figure out which are illegitimate connections
  - Send RST for the illegitimate connections (which closes the connection)
  - Does not require protocol stack modification
  - Must promptly and reliably distinguish bad addresses from good addresses
Ping of Death Attack

- Attacker sends a flood of pings to the victim
  - Victim’s bandwidth is saturated, as victim is unable to keep up with the flood
- Attack requires attacker’s bandwidth to be greater than the victim’s bandwidth
  - Attacker’s bandwidth is limited by the smallest bandwidth on the attack route
  - If it is smaller than the victim’s bandwidth then the attack fails
Teardrop Attack

- Recall: Fragmentation of IP packets
  - Broken into smaller pieces
  - Require re-assembly
- Attack exploits bugs in re-assembly code
- Attacker sends malformed fragments
  - Overlapping (e.g., second fragment contained in first fragment)
  - Oversized
  - …
  - Unexpected conditions, can cause crash
Low-rate DoS Attack

- Attack throttles TCP flows
  - Brings down to a small fraction of normal rate, in a hard to detect manner

- Attack relies on TCP’s congestion control
  - TCP congestion control is very robust, but relies on implicit assumptions of cooperation
  - Exploited in high-rate attacks (e.g., flooding)
  - Detection mechanisms rely on the high rate
  - Low-rate attacks elude such detection
Low-rate DoS Attack (cont’d)

- Attacker maintains a low average rate, but sends *high-rate bursts* for very brief periods of time
  - A burst tricks TCP to respond as if high congestion (and reduce throughput)
  - Attacker maintains the reduced throughput situation with periodic bursts (yet low average rate)
- Hard to detect
Low-rate DoS Attack Mitigation

• Deterministic TCP congestion control behavior makes attack easier
  – Attacker can exploit TCP’s retransmission timeout mechanism
  – Attacker can time the bursts for optimal damage (periodic bursts of chosen period)

• Can be mitigated through randomization of TCP’s retransmission timeout parameters
  – But not eliminated without considerable performance sacrifice (weakness is inherent)
Traffic Hijacking

• Corruption of dynamic routing tables
  – A node lies, so as to become “next hop” to destination X on other nodes’ routing tables

• In observed attacks, multiple nodes worked in a coordinated manner
  – Diverted huge North-American traffic through foreign countries
  – Traffic was surely observed, possibly modified (“person in the middle attack”)
Packet Drop, Unintentional DoS

- Packet drop attacks
  - Blackhole attack – router drops all packets (ends up being removed from routing tables)
  - Grayhole attack – router selectively and/or intermittently drops packets

- Unintentional DoS, e.g.,
  - Link from a high-traffic site to a low-traffic one
  - If name is a short edit distance away from the name of a high-traffic site (e.g., utube)
Forged TCP Resets

- Flip the “reset” flag in TCP header from 0 to 1
- Kills the connection (all its traffic is discarded)
- Often (but not always) an attack
  - Can be useful for a defensive tool against suspicious connections
- Was used (until 2008) by ISPs to cripple certain applications (e.g., peer-to-peer)
  - FCC ordered a stop to the practice
DNS Hijacking

- Subversion of DNS queries
- Done by attackers (e.g., for phishing where you’re sent to a fake version of your bank’s web site)
- Done by ISPs
  - For collecting statistics
  - For displaying their choice of ads (for which they collect money)
  - Bad for customers, can break other functionality
Pharming

• Redirection of a website’s traffic to a fake site
• Can happen in many ways, e.g.,
  – Malware overwrites local “hosts” file (that has IP addresses, bypassing the need for DNS)
  – Router compromise, e.g., malware-ridden firmware, or overwriting in the router the “trusted DNS server” entry with a rogue one under the control of the attacker (the router’s entry takes precedence over the ISP-suggested DNS server)
DoS vulnerability caused by session hijacking

• Suppose attacker can guess seq. number for an existing connection:
  – Attacker can send Reset packet to close connection. Results in DoS.
  – Naively, success prob. is $1/2^{32}$ (32-bit seq. #'s).
  – Most systems allow for a large window of acceptable seq. #'s
    • Much higher success probability.
• Attack is most effective against long lived connections, e.g. BGP.
## Categories of Denial-of-service Attacks

<table>
<thead>
<tr>
<th>Locally</th>
<th>Stopping services</th>
<th>Exhausting resources</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Process killing</td>
<td>• Spawning processes to fill the process table</td>
</tr>
<tr>
<td></td>
<td>• Process crashing</td>
<td>• Filling up the whole file system</td>
</tr>
<tr>
<td></td>
<td>• System reconfiguration</td>
<td>• Saturate comm bandwidth</td>
</tr>
<tr>
<td>Remotely</td>
<td>• Malformed packets to crash buggy services</td>
<td>• Packet floods (Smurf, SYN flood, DDoS, etc)</td>
</tr>
</tbody>
</table>

- Stopping services
  - Process killing
  - Process crashing
  - System reconfiguration

- Exhausting resources
  - Spawning processes to fill the process table
  - Filling up the whole file system
  - Saturate comm bandwidth
Internet Control Message Protocol

• Provides feedback about network operation
  – Error reporting
  – Reachability testing
  – Congestion Control

• Example message types
  – Destination unreachable
  – Time-to-live exceeded
  – Parameter problem
  – Redirect to better gateway
  – Echo/echo reply - reachability test
  – Timestamp request/reply - measure transit delay
Distributed DoS (DDoS)
Hiding DDoS Attacks

- Reflection
  - Find big sites with lots of resources, send packets with spoofed source address, response to victim
    - PING => PING response
    - SYN => SYN-ACK

- Pulsing zombie floods
  - each zombie active briefly, then goes dormant;
  - zombies taking turns attacking
  - making tracing difficult
Cryptographic network protection

• Solutions above the transport layer
  – Examples: SSL and SSH
  – Protect against session hijacking and injected data
  – Do not protect against denial-of-service attacks caused by spoofed packets

• Solutions at network layer
  – Use cryptographically random ISNs [RFC 1948]
  – More generally: IPsec
  – Can protect against
    • session hijacking and injection of data
    • denial-of-service attacks using session resets