Computer Security and Penetration Testing

TCP/IP Vulnerabilities
Objectives

- Give a definition of TCP/IP
- Know the steps of TCP/IP communication
- Recognize weaknesses in TCP/IP
- Identify steps in protecting information from vulnerabilities in TCP/IP
TCP/IP Vulnerabilities

• Transmission Control Protocol/Internet Protocol (TCP/IP)
  – Suite of protocols that underlie the Internet
  – Comprises many protocols and applications
  – Common language of networked computers
  – Makes transferring information fast and efficient
• IP has tools to correctly rout packets
• TCP is responsible for safe and reliable data transfer between host computers
TCP/IP Vulnerabilities (continued)

- Illegitimate users take advantage of TCP/IP vulnerabilities
  - By exploiting the “three-way handshake”
- Unauthorized users may launch a denial-of-service attack on the destination computer
  - Floods network with so many additional requests that regular traffic is slowed or completely interrupted
TCP/IP Vulnerabilities (continued)

![TCP/IP Diagram](image)

**Figure 5-1** TCP/IP
Data Encapsulation

• **Data encapsulation**
  – Enclosing higher-level protocol information in lower-level protocol information
  – Also called data hiding
  – Implementation details of a class are hidden from user
Data Encapsulation (continued)

<table>
<thead>
<tr>
<th>Data Link Layer</th>
<th>Sent out to gateway router and on to described recipient</th>
<th>Frame (Ethernet) Header IP Header TCP Header Application Header Data.doc Frame (Ethernet) Footer</th>
<th>Frame (Ethernet) Header IP Header TCP Header Application Header Data.doc Frame (Ethernet) Footer</th>
<th>Received from gateway router and checked as to whether this is proper recipient</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Layer</td>
<td>Prepared for transfer and loaded to NIC queue</td>
<td>IP Header TCP Header Application Header Data.doc</td>
<td>IP Header TCP Header Application Header Data.doc</td>
<td>Frame header and trailer stripped, then packets passed to TCP Transport Layer</td>
</tr>
<tr>
<td>Transport Layer</td>
<td>Divided into packets</td>
<td>TCP Header Application Header Data.doc</td>
<td>TCP Header Application Header Data.doc</td>
<td>Packets checked for data-integrity and reassembled in proper order, then TCP Headers stripped</td>
</tr>
<tr>
<td>Application Layer</td>
<td>Saved as a document type</td>
<td>Application Header Data.doc</td>
<td>Application Header Data.doc</td>
<td>Verified as a document of specified type, and opened by that application</td>
</tr>
<tr>
<td>User Data</td>
<td>the contents of a document</td>
<td>Data</td>
<td>Data</td>
<td>Read by recipient</td>
</tr>
<tr>
<td>Layers</td>
<td>Process</td>
<td>Image</td>
<td>Image</td>
<td>Process</td>
</tr>
</tbody>
</table>

**Figure 5-2** Data encapsulation
IP (Internet Protocol)

- Internet Protocol (IP)
  - Transmits data from source to final destination
  - Network protocol operating at layer 3 of the OSI Model
    - And layer 2 or 3 of the TCP/IP Model
  - IP is connectionless
    - No guarantee of delivery of packets to the destination
- IP routes packets over network hardware
IP (Internet Protocol) (continued)

• IP addresses formats
  – IPv4 (32-bit address)
    • Usually written as a dotted-decimal, e.g., 192.168.100
  – IPv6 (128-bit address)
    • Usually written as eight groups of four hex digits, e.g.,
      2001:0db8:85a3:08d3:1319:8a2e:0370:7334

• **IP address exhaustion** date
  – Approximately the beginning of 2011
IP (Internet Protocol) (continued)

- IP packets often arrive out of sequence
  - Vulnerability that attackers can exploit
- When a large IP packet is sent over a network, it is broken down
  - Called fragmentation
IP (Internet Protocol) (continued)

<table>
<thead>
<tr>
<th>Table 5-1 IP header</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Version</strong></td>
</tr>
<tr>
<td>Identification</td>
</tr>
<tr>
<td>Time to Live</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source Address</th>
<th>Destination Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Options</td>
<td>Padding</td>
</tr>
<tr>
<td>TCP header, followed by data</td>
<td></td>
</tr>
</tbody>
</table>
### Table 5-2  Components of the IP header

<table>
<thead>
<tr>
<th>Fields within an IP Header</th>
<th>Size in Bits</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>4</td>
<td>Set to 4 to specify IPv4 packets or to 6 to specify IPv6 packets.</td>
</tr>
<tr>
<td>IHL</td>
<td>4</td>
<td>Internet Header Length (IHL) is the length of the Internet header; normal value=5.</td>
</tr>
<tr>
<td>Type of Service</td>
<td>8</td>
<td>Points out the quality of the requested service; the quality of service specifies trade-offs between low delay, high reliability, and high throughput, and specifies the precedence level of a packet.</td>
</tr>
<tr>
<td>Fields within an IP Header</td>
<td>Size in Bits</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Total Length</td>
<td>16</td>
<td>Specifies the total length of the datagram packet that is being relayed; because this is a 16-bit field, the maximum size of the IP datagram is 65,535 bytes.</td>
</tr>
<tr>
<td>Identification</td>
<td>16</td>
<td>Specifies an identity assigned to all of the packets in a datagram so that if the datagram is fragmented, it can be reassembled at the destination.</td>
</tr>
<tr>
<td>Flags</td>
<td>3</td>
<td>Three flags are used in the fragmentation of the datagram. The first is reserves. The second is Don’t Fragment (DF), which is used for testing purposes, and is ignored by most higher-level protocols. Setting DF to 1 means the packet should not be fragmented. The third flag is More Fragments (MF). If MF is set to 0, it is the last piece. If set to 1, it will be followed by others.</td>
</tr>
<tr>
<td>Fragment Offset</td>
<td>13</td>
<td>Specifies the sequence of the fragment in the datagram.</td>
</tr>
<tr>
<td>Time to Live</td>
<td>8</td>
<td>Specifies the maximum time the datagram can remain on the network, stated in terms of how many router hops it can make.</td>
</tr>
<tr>
<td>Protocol</td>
<td>8</td>
<td>Specifies the next-level protocol to which the datagram belongs.</td>
</tr>
<tr>
<td>Header Checksum</td>
<td>16</td>
<td>Specifies the checksum on only the datagram header. It is a simple 16-bit checksum calculated by dividing the header bytes into words (a word is two bytes) and then adding them together. It is needed because the header changes when the value in the Time to Live field changes. At that point, it is recalculated. Each router performs this checksum, and if the calculated figure and the contents of the Header Checksum field are found to be dissimilar, then that router discards the packet as corrupted.</td>
</tr>
<tr>
<td>Source Address</td>
<td>32</td>
<td>Specifies the source IP address of the datagram; this is always the original sender, even if the intervening sender was a router or bridge.</td>
</tr>
<tr>
<td>Destination Address</td>
<td>32</td>
<td>Specifies the destination IP address of the datagram; this is always the final destination of the packet, even if the intervening destination is a router or a bridge.</td>
</tr>
<tr>
<td>Options</td>
<td>Varies</td>
<td>May or may not be present in a datagram; these options are related to routing packets over the network.</td>
</tr>
<tr>
<td>Padding</td>
<td>Varies</td>
<td>Adds extra “0” characters to pad out the header to a multiple of 32 bits.</td>
</tr>
</tbody>
</table>
TCP

• Uses a connection-oriented design
  – Participants in a TCP session must create connection
• Connection is called the three-way handshake
• Provides connection-oriented services between a source and destination computer
  – And guarantees delivery of packets
• Packets reach the application layer in the right order
  – TCP identifies and assembles packets based on sequence numbers
TCP (continued)

• Source and destination computers exchange the initial sequence number (ISN)
  – When a connection is made
• Packets are accepted within a particular range
  – Specified during the establishment of a connection
TCP (continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>16-bit source port number</td>
<td>16-bit destination port number</td>
</tr>
<tr>
<td>32-bit sequence number</td>
<td></td>
</tr>
<tr>
<td>32-bit acknowledgement number</td>
<td></td>
</tr>
<tr>
<td>4-bit header</td>
<td>Reserved (6-bits)</td>
</tr>
<tr>
<td></td>
<td>U A P R S F</td>
</tr>
<tr>
<td></td>
<td>R C S S Y I</td>
</tr>
<tr>
<td></td>
<td>G K H T N</td>
</tr>
<tr>
<td>16-bit TCP checksum</td>
<td>16-bit window size</td>
</tr>
<tr>
<td></td>
<td>Options (if any) plus Padding</td>
</tr>
<tr>
<td></td>
<td>Data (if any)</td>
</tr>
<tr>
<td>16-bit urgent pointer</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 5-3** TCP header
TCP (continued)

<table>
<thead>
<tr>
<th>Fields in a TCP Header</th>
<th>Size in Bits</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source port number</td>
<td>16</td>
<td>Can be any port number from 1 to 65,535; there is no reason for a source port for a Web-page request to be port 80.</td>
</tr>
<tr>
<td>Destination port number</td>
<td>16</td>
<td>The port number of the destination; can be any port number from 1 to 65,535.</td>
</tr>
<tr>
<td>Sequence number (ISN)</td>
<td>32</td>
<td>Sequence number of the first data octet in this segment.</td>
</tr>
<tr>
<td>Acknowledgment number</td>
<td>32</td>
<td>If the ACK bit is set, this field contains the value of the next sequence number that is expected by the sender.</td>
</tr>
<tr>
<td>Header</td>
<td>4</td>
<td>Offset where data begins in the packet.</td>
</tr>
<tr>
<td>Reserved</td>
<td>6</td>
<td>Reserved for future use.</td>
</tr>
<tr>
<td>URG</td>
<td>1</td>
<td>Urgent pointer field.</td>
</tr>
<tr>
<td>ACK</td>
<td>1</td>
<td>Acknowledgment field.</td>
</tr>
<tr>
<td>PSH</td>
<td>1</td>
<td>Push Function.</td>
</tr>
<tr>
<td>RST</td>
<td>1</td>
<td>Reset the connection.</td>
</tr>
</tbody>
</table>
### Table 5-3 Components of the TCP header (continued)

<table>
<thead>
<tr>
<th>Fields in a TCP Header</th>
<th>Size in Bits</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYN</td>
<td>1</td>
<td>Synchronize sequence numbers.</td>
</tr>
<tr>
<td>FIN</td>
<td>1</td>
<td>No more data from the sender.</td>
</tr>
<tr>
<td>Window size</td>
<td>16</td>
<td>Number of data octets starting from one.</td>
</tr>
<tr>
<td>TCP checksum</td>
<td>16</td>
<td>Checksum for error-checking of the header and data in the packet.</td>
</tr>
<tr>
<td>Urgent pointer</td>
<td>16</td>
<td>When the URG bit is set, this field is interpreted; it contains the value of the urgent pointer, which is higher than the sequence number in this segment.</td>
</tr>
<tr>
<td>Options (if any)</td>
<td>Varies</td>
<td>Routing options for the packet.</td>
</tr>
<tr>
<td>Padding</td>
<td>Varies</td>
<td>Pads the header to achieve a multiple of 32 bits.</td>
</tr>
<tr>
<td>Data (if any)</td>
<td>Varies</td>
<td>Data to be transmitted.</td>
</tr>
</tbody>
</table>
Connection Setup and Release

• Three-way handshake sets up and releases a connection

• **TCP packet flags:** URG, ACK, PSH, RST, SYN, and FIN

• Packets can have more than one flag set
  – Normally a packet will have only one flag sent, except with SYN/ACK or FIN/ACK

• Three packets in a TCP connection:
  SYN --> SYN/ACK --> ACK
Connection Setup and Release (continued)

• Connection Setup
  – Source computer delivers a SYN packet to the destination computer
    • Packet has the initial sequence number (ISN)
    • ISN is indicated by whether the SYN bit is “set”
  – Receiving computer transmits a SYN with an acknowledgment, ACK
  – Source computer sends an ACK to the destination computer as a response
    • With an “in-range” sequence number
Figure 5-4  Connection setup
Connection Setup and Release (continued)

• Connection Release
  – Source computer sends a FIN packet to the destination computer
  – Destination computer then sends a FIN/ACK packet
  – Source computer sends an ACK packet
  – Either computer could send an RST and close the session (reset) immediately
TCP Timers

- All TCP sessions are tracked with timers built into the TCP protocol
- Timers used by TCP/IP
  - Connection establishment
    - A session will not be established if it takes longer than 75 seconds for the destination server to respond
  - FIN_WAIT
    - Waits for FIN packets. Its default value is 10 minutes
TCP Timers (continued)

• Timers used by TCP/IP (continued)
  – TIME_WAIT
    • Default value for this timer is two minutes
    • Waits for packets to arrive at the destination computer
  – KEEP_ALIVE
    • Checks to see if the destination computer is active
    • Computer may send a test packet every two hours to verify whether the other computer is alive and idle
Vulnerabilities in TCP/IP

• During the development of TCP/IP in the 1980s
  – Security was not a priority
• Since 1990, security has become a serious problem
• Some of the vulnerabilities
  – IP spoofing
  – Connection hijacking
  – ICMP attacks
  – TCP SYN attacks
  – RIP attacks
IP Spoofing

• Steps
  – Attackers send packets to the victim or target computer with a false source address
  – Victim accepts the packet and sends a response “back” to the indicated source computer
  – Attacker must guess the proper sequence numbers to send the final ACK packet
• Hacker may have a connection to victim’s machine
  – And hold it as long as the computer remains active
IP Spoofing (continued)

• Sequence Guessing
  – Hacker sends a few connections to the victim
    • Learns how quickly sequence number is incrementing
  – Attacker then sends a spoofed ACK packet with a “best guess” victim’s sequence number
  – Hacker can guess the sequence number because the number is generated using a global counter
    • And is incremented in fixed units
IP Spoofing (continued)

• Source Routing
  – Sender using **source routing** can specify return path
    • Through which the destination computer sends its reply
  – Attacker looks for an intermediate computer or router
    • That could forward packets to the target computer
  – Most newer routers and firewalls are configured to drop source-routed packets
Connection Hijacking

• **Connection hijacking**
  – Allows an attacker to control an existing connection

• **Steps**
  – An attacker desynchronizes a series of packets between the source and destination computer
  – Extra packets sent to one of the victims force the victim to choose which packet to accept
  – If the victim chooses to discard the authentic packets and interacts with the spoofed packets
    • The attacker has hijacked the connections
ICMP Attacks

- Packets are used to send fraudulent or deceptive connection information among computers
- ICMP is used to test for connectivity using utilities such as the `ping command`
- Denial-of-service (DoS) attacks can be formulated by using ICMP packets
  - Destination Unreachable and Time to Live Exceeded
- Attackers transmitting spoofed packets can successfully reset existing connections
TCP SYN Attacks

- Exploits host implementation of three-way handshake
- When Host B receives the SYN request from A, it must keep track of the partially opened connection
  - In a queue for at least 75 seconds
- Most systems are limited and can keep track of only a small number of connections
- An attacker can overflow the listen queue by sending more SYN requests than the queue can handle
  - SYN flooding
RIP Attacks

• Take advantage of RIP (Routing Information Protocol)
• RIP
  – Essential component in a TCP/IP network
  – Distribution of routing information within networks
• RIP packet is often used without verification
  – Attacks on RIP change the destination of data
• Once the router is modified, it transmits all of the packets to the hacker computer
Securing TCP/IP

- Data in packets is not encrypted or authenticated
- Packet sniffer can observe contents of the packets
- Attackers can send spoofed packets from any computer
- Must employ many methods simultaneously to achieve success in this area
Securing TCP/IP (continued)

• Methods to decrease vulnerabilities in TCP/IP
  – Modify default timer values
  – Increase the number of simultaneous connections that a computer can handle
  – Reduce the time limit used to listen for replies to the SYN/ACK in the three-way handshake
  – Change method used to generate sequence numbers
  – Firewall rules that block spoofed packets
Securing TCP/IP (continued)

• Methods to decrease vulnerabilities in TCP/IP (continued)
  – Avoid using the source address authentication
  – If an operator allows outside connections from trusted hosts, enable encryption sessions at the router
  – Packets can be encrypted or sent via encrypted VPN
IP Security Architecture (IPSec)

• **IP Security Architecture (IPSec)**
  – Collection of Internet Engineering Task Force (IETF) standards
  – Defines an architecture at the Internet Protocol (IP) layer that protects IP traffic
    • By using various security services
Table 5-4  Some IPSec protocols

<table>
<thead>
<tr>
<th>RFC Number</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2401</td>
<td>Security Architecture for the Internet Protocol</td>
<td>The main IPSec document, describing the architecture and general operation of the technology, and showing how the different components fit together</td>
</tr>
<tr>
<td>2402</td>
<td>IP Authentication Header</td>
<td>Defines the IPSec Authentication Header (AH) protocol used for ensuring data integrity and origin verification</td>
</tr>
<tr>
<td>2403</td>
<td>The Use of HMAC-MD5-96 within ESP and AH</td>
<td>Describes a particular encryption algorithm for use by AH and ESP called Message Digest 5 (MD5)</td>
</tr>
<tr>
<td>2404</td>
<td>The Use of HMAC-SHA-1-96 within ESP and AH</td>
<td>Describes a particular encryption algorithm for use by AH and ESP called Secure Hash Algorithm 1 (SHA-1)</td>
</tr>
</tbody>
</table>
### IP Security Architecture (IPSec) (continued)

**Table 5-4  Some IPSec protocols (continued)**

<table>
<thead>
<tr>
<th>RFC Number</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2406</td>
<td>IP Encapsulating Security Payload (ESP)</td>
<td>Describes the IPSec Encapsulation Security Payload (ESP) protocol that provides data encryption for confidentiality</td>
</tr>
<tr>
<td>2408</td>
<td>Internet Security Association and Key Manage-</td>
<td>Defines methods for exchanging keys and negotiating security associations</td>
</tr>
<tr>
<td></td>
<td>ment Protocol (ISAKMP)</td>
<td></td>
</tr>
<tr>
<td>2409</td>
<td>The Internet Key Exchange (IKE)</td>
<td>Describes the Internet Key Exchange (IKE) protocol used to negotiate security associations and exchange keys between devices for secure communications; based on ISAKMP and OAKLEY</td>
</tr>
<tr>
<td>2412</td>
<td>The OAKLEY Key Determination Protocol</td>
<td>Describes a generic protocol for key exchange</td>
</tr>
</tbody>
</table>