AN TOÀN LỚP 4: TCP/IP ATTACKS

NGUYEN HONG SON
PTITHCM

Introduction (1/2)

- TCP provides a full duplex reliable stream connection between two end points
- A connection is uniquely defined by the quadruple (IP address of sender, TCP port number of the sender, IP address of the receiver, TCP port number of the receiver)
- Every byte that is sent by a host is marked with a sequence number (32 bits integer) and is acknowledged by the receiver using this sequence number
- The sequence number for the first byte sent is computed during the connection opening. It changes for any new connection based on rules designed to avoid reuse of the same sequence number for two different sessions of a TCP connection

Introduction (2/2)

- There are a number of serious security flaws inherent in the protocols
- Flaws even arise due to the bad implementation and improper configuration of the applications using these protocol suites
- The attacks are classified into Active attacks and passive attacks depending on the behaviour of the attacker
- A passive attack (like sniffer) is one that can take place by eavedropping.
- An active attack (like Hijacking) is one that requires interaction, such as injecting something into the data stream or change, delete, reroute, add, forge or divert data

TCP Operations

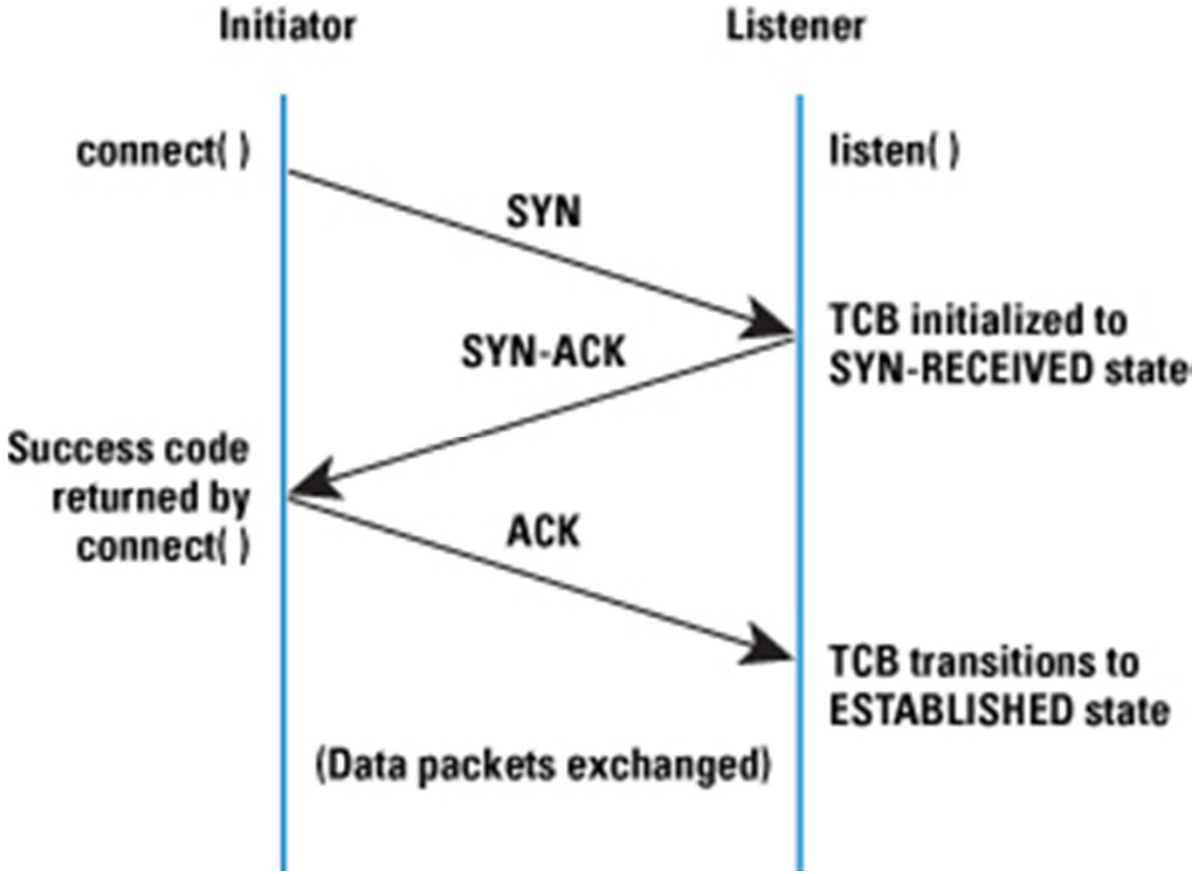
- Connection setup
- Data transfer
- Connection close

TCP Connection Setup: Three-Way Handshake

1.The valid user initiates a connection with the server. This is accomplished by the valid user sending a packet to the server with the SYN bit set and the user's initial Sequence Number (ISN).

2.The server receives this packet and sends back a packet with the SYN bit set and an ISBN for the server, plus the ACK bit set identifying the user's ISN incremented by a value of one.

3.The valid user acknowledges the server by returning a packet with the ACK bit set and incrementing the servers ISN by one.


Initiator
Listener
connect( )
listen( )
SYN
TCB initialized to
SYN-RECEIVED state
SYN-ACK
Success code
returned by
connect( )
ACK
TCB transitions to
ESTABLISHED state
(Data packets exchanged)

CLIENT

SERVER

SYN <Clt ISN><WIN>

SYN <Svr ISN><WIN> / ACK (Clt ISN + 1)

ACK (Svr ISN + 1)

Data Transfer

- Sequence number
- ACK number

TCP Connection Close

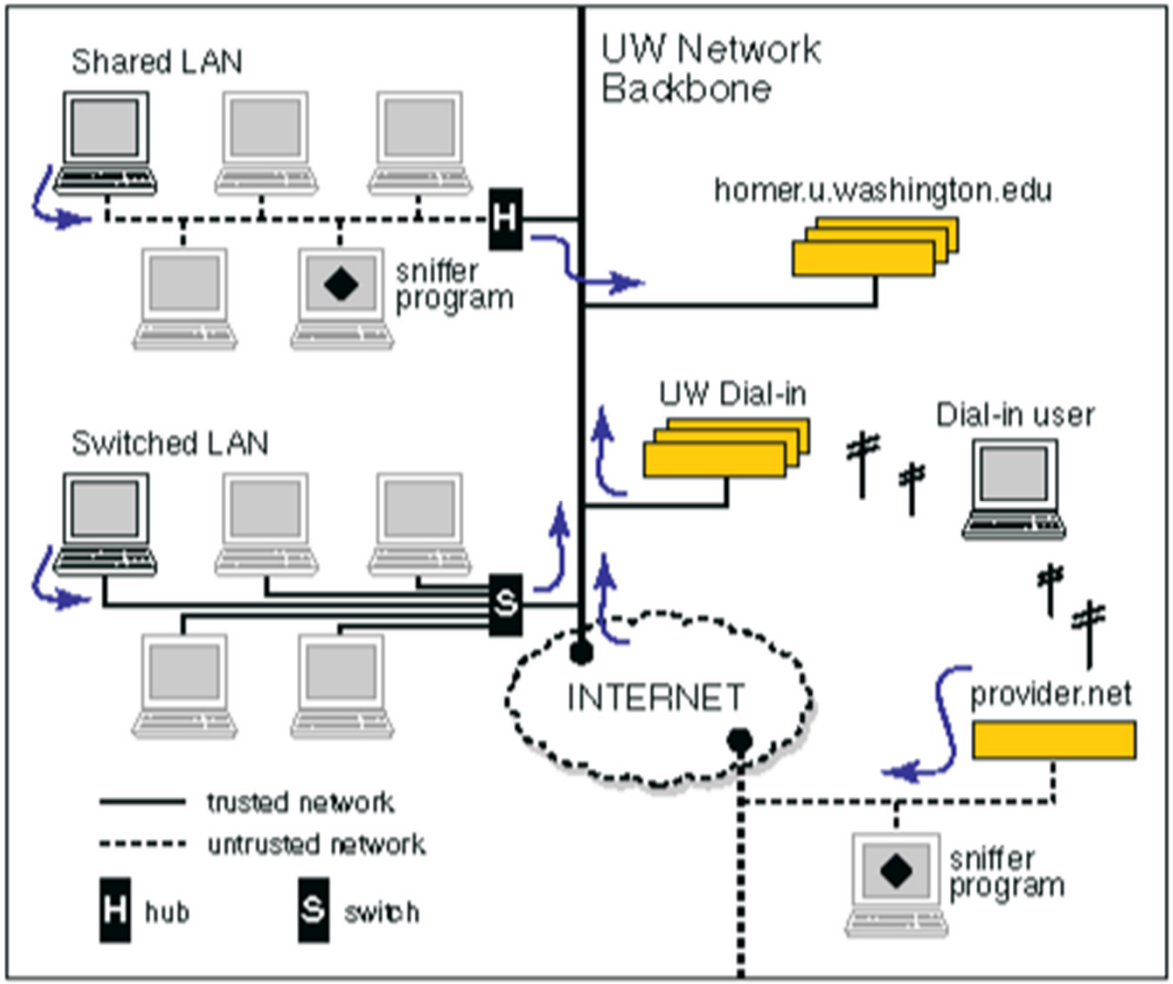
- Connection can be closed from either side due to a timeout, or upon receipt of a package with the FIN or RST flag set
- Upon receipt of a packet with the RST flag set, the receiving system closes the connection, and any incoming packets for the session are discarded
- If the FIN flag is set in a packet, then the receiving system goes through the process of closing the connection, and any packets received while closing the connection are still processed

Passive Attacks

- Sniffing
- Protocols vulnerable to sniffing?
- Methods for sniffing
- Sniffing Prevention
- Detect packet sniffers

Sniffing

- The act of intercepting and reading any or all network traffic that is being transmitted across a shared network communication channel.
- Sniffing programs are of 2 two forms:
  - Commercial packet sniffers are used to help maintain networks.
  - Underground packet sniffers are used to break into computers.


Shared LAN
UW Network
Backbone
homer.u.washington.edu
H
sniffer
program
UW Dial-in
Dial-in user
Switched LAN
S
INTERNET
provider.net
trusted network
untrusted network
H hub
S switch
sniffer
program

Protocols vulnerable to sniffing?

- Telnet and rlogin : Sniffing can capture the keystrokes as the user types them, including the user name and password.
- HTTP :The default version of HTTP has numerous holes. Many web sites use "Basic" authentication, which sends passwords across the wire in plain-text. Many web sites use another technique which prompts the user for a username and password, which are also sent across the network in plaintext. Data sent in clear-text.
- SNMP: Almost all SNMP traffic is SNMPv1, which has no good security. SNMP passwords (called community-strings) are sent across the wire in the clear.
- NNTP, POP, IMAP, FTP, : Passwords sent in the clear. Data sent in clear

Methods for sniffing

- Both the hardware and the software components of the NIC need to provide mechanisms for capturing the raw packets.
- After the network traffic is processed by the NIC, software mechanisms are needed to filter the captured data.
- Finally, a mechanism is required to extract and reconstruct the data portion of the captured packets, and to display what you get in a readable format
- Sniffing programs : **wireshark, dsniff, ettercap, sniffit, hunt, lcrzoex**
- A sniffer program makes the network interface card (NIC) on the machine enter into a so-called promiscuous mode

Sniffing Prevention (1/2)

- Authentication schemes such as MD4 and MD5, KERBEROS, DESLOGIN, s/key, and SSH are available to prevent the clear text transmission of user names and passwords across a network.
- Public key encryption programs such as PGP are available to encrypt electronic mail (E-mail) to prevent the contents from being read.
- Sniffer programs running in promiscous mode can be found out by identifying the sessions currently running on the machine
- Often a sniffer log becomes so large that the file space is all used up. On a high volume network, a sniffer will create a large load on the machine. These sometimes trigger enough alarms that the administrator will discover a sniffer.

Sniffing Prevention (2/2)

- To detect a sniffing device that only collects data and does not respond to any of the information, requires physically checking all your ethernet connections by walking around and checking the ethernet connections individually
- Using interfaces that will not allow processes to run in promiscuous mode
- Using cctive hubs send to each system only packets intended for it

Detection of Sniffers

- Physically checking all your Ethernet connections by walking around.
- Local host: Observing the output of ifconfig -a or equivalent (which includes the word PROMISC)
- DNS Test
- Ping Test
- ICMP Ping Latency Test
- ARP Test
- Using IDS

Programs to Detect Sniffers

- **Anti Sniff**: It has the ability to monitor a network and detect if a computer is in promiscuous mode. http://www.securitysoftwaretech.com/antisniff/download.html
- **Neped**: It detects network cards on the network that are in promiscuous mode by exploiting a flaw in the ARP protocol as implemented on Linux machines. ftp://apostols.org/AposTools/snapshots/neped/neped.c
- **ARP Watch**: ARPWatch keeps track of Ethernet/IP address pairings. This is useful when you suspect you are being arp-spoofed. : ftp://ftp.ee.lbl.gov/arpwatch.tar.Z
- **Snort**: Snort is an excellent Intrusion Detection System and its arp-spoof preprocessor can be used to detect instances of ARP Spoofing http://www.snort.org/

Active Attacks

- IP spoofing
- Hijacking
- Sequence Number Guessing
- Syn Flooding

IP spoofing (1/2)

- In spoofing attack, the hacker performs sniffing and listens to traffic as it's passed along the network from sender to receiver. The hacker then uses the information gathered to spoof or uses an address of a egitimate system
- IP spoofing: To gain access, intruders create packets with spoofed source IP addresses
- This exploits applications that use authentication based on IP addresses and leads to unauthorized user and possibly root access on the targeted system

IP spoofing (2/2)

Spoofing is classified into

- Non-blind spoofing: Using the spoofing to interfere with a connection that sends packets along your subnet.
- Blind spoofing: Using the spoofing to interfere with a connection (or creating one), that does not send packets along your cable.

Hijacking (1/2)

- Once the intruders have root access on a system (by IP spoofing), they can hijack existing terminal and login connections from any user on the system
- In taking over the existing connections, intruders can bypass one-time passwords and other strong authentication schemes by tapping the connection after the authentication is complete
- A legitimate user connects to a remote site through a login or terminal session; the intruder hijacks the connection after the user has completed the authentication to the remote location; the remote site is now compromised

Hijacking (2/2)

- Session hijacking involves the following three steps to perpetuate an attack:
- **Tracking the session:** The hacker identifies an open session and predicts the sequence number of the next packet.
- **Desynchronizing the connection:** The hacker sends the valid user's system a TCP reset (RST) or finish (FIN) packet to cause them to close their session.
- **Injecting the attacker's packet:** The hacker sends the server a TCP packet with the predicted sequence number, and the server accepts it as the valid user's next packet.

TCP attacks: Connection Killing

- **Setup**

host A <------X----------->host B | A,B have a TCP connection running

host S <------/ A,S on same subnet

- **Using reset (RST):** To be accepted, only the sequence number has to be correct (no ACK in a RST packet). Calculate (from B's packets) the sequence number for A's packets (from B's ACK's), and fire a bogus RST packet from S (faking to be A) to B
- **Using close a connection (FIN):** sniff the current SEQ/ACK of the connection we can pretend to be either host A or B, and provide the other host with CORRECT packetinformation, and an evil FIN flag.

TCP Attack: Connection Hijacking

- **Setup**

host A <------X----------->host B | A,B have a TCP connection running (example A Telnet into B)

host S <------/ A,S on same subnet

- TCP separates good and bogus packets by their SEQ/ACK numbers. B trusts the packets from A because of its correct SEQ/ACK numbers
- If mess up A's SEQ/ACK, B would stop believing A's real packets. Then, could impersonate to be A, but using correct SEQ/ACK numbers. Now, have taken over the connection. This is called 'Hijacking' a connection

TCP Attack: Connection Hijacking

- To mess up A's SEQ/ACK numbers simply insert a data packet into the stream at the right time (S as A->B), the server B would accept this data, and update ACK numbers, A would continue to send it's old SEQ numbers, as it's unaware of our spoofed data.
- What dangerous things with connection hijacking

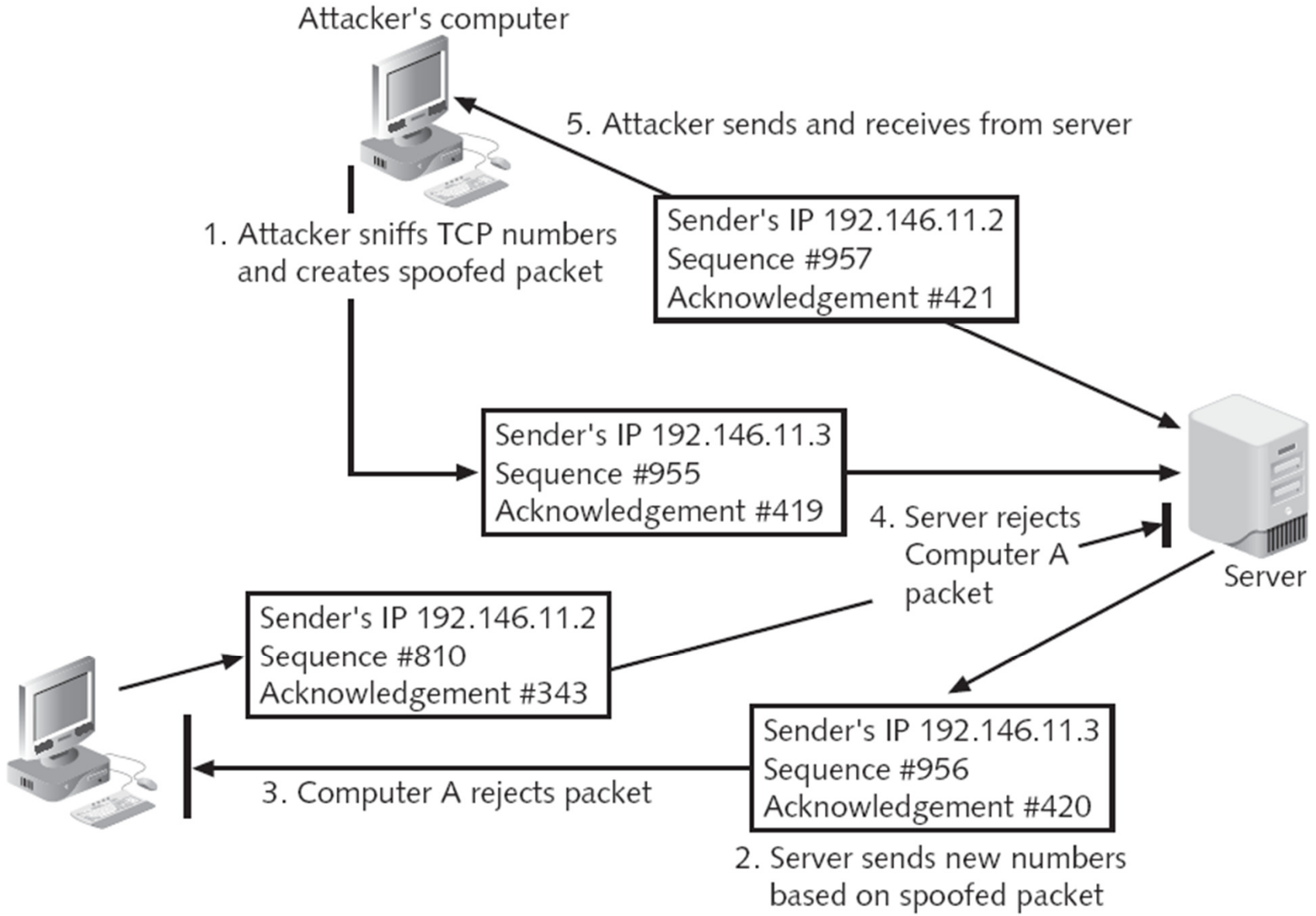


Figure 4-12 TCP/IP hijacking

TCP Attacks: Sequence Number Guessing (1/2)

- If TCP sequence numbers are predictable, a hacker can forge a connection from another machine
- In order to successfully perform a TCP sequence prediction attack, the hacker must sniff the traffic between two systems.
- Next, hacking tool must successfully guess the sequence number or locate an ISN to calculate the next sequence number. This can be more difficult than it sounds, because packets travel very fast.
- The hacking tools issue packets with the sequence numbers that the target system is expecting. But the hacker's packets must arrive before the packets from the trusted system whose connection is being hijacked. This is accomplished by flooding the trusted system with packets or sending a RST packet to the trusted system so that it is unavailable to send packets to the target system.

TCP Attacks: Sequence Number Guessing (2/2)

- The initial sequence numbers are intended to be more or less random. RFC 793 specifies that the 32-bit counter be incremented by 1 in the low-order position about every 4 microseconds
- Some unix versions like Free-BSD increment it by a constant every second, and by another constant for each new connection

Prevent Session Hijacking (1/2)

- Prevent sniffing: The most effective protection is encryption, such as Internet Protocol Security (IPSec) Secure Shell (SSH, an encrypted Telnet) and Secure Sockets Layer (SSL, for HTTPS traffic).
- Reducing the potential methods of gaining access to your network—for example, by eliminating remote access to internal systems
- Using VPN for remtote access (if need)

Prevent Session Hijacking (2/2)

- Newer operating systems have attempted to secure themselves from session hijacking by using pseudorandom number generators to calculate the ISN, making the sequence number harder to guess
- But, random leads to protocol problems like duplicate packets and reincarnations of packets of the old connection at the server, due to which the server will not be able to distinguish if the packets were from the current session or from the previous connection
- One way to avoid this is to allot sequence number space to each port, and the sequence numbers are incremented according to the following relationship

$$ISN = M + F(localhost, localport, remotehost, remoteport).$$

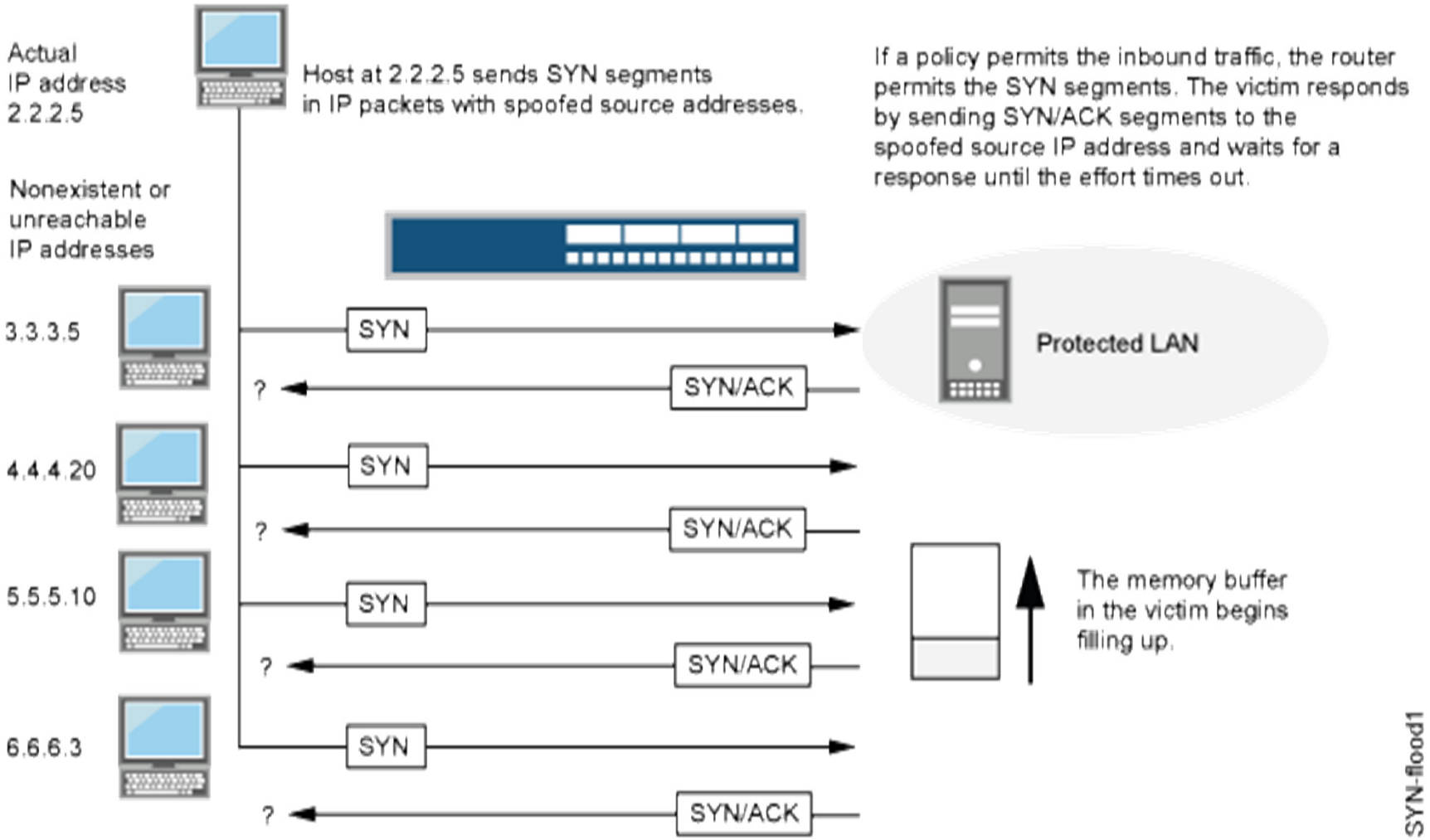
However,

this security measure is ineffective if the attacker is able to sniff packets

SYN Flooding (1/2)

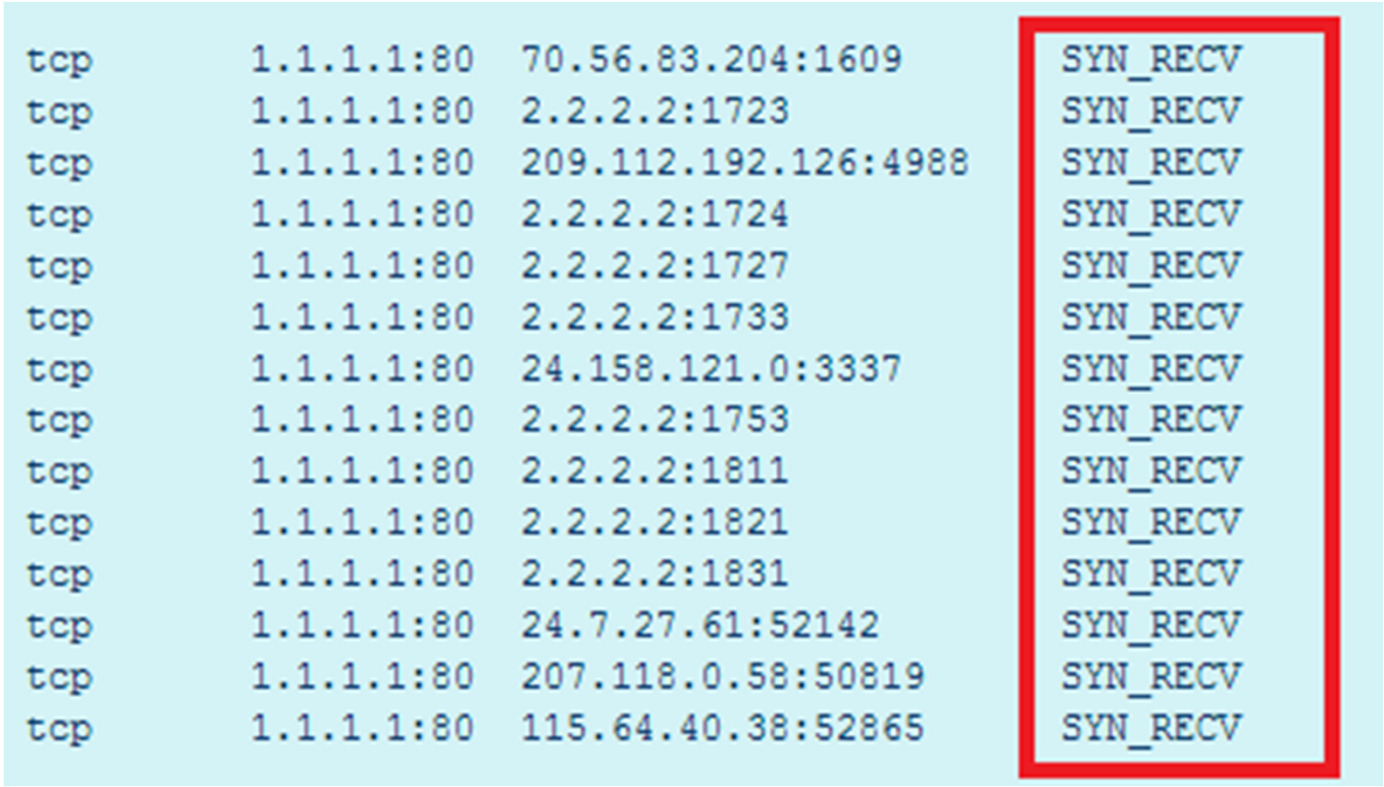
- A SYN packet is the first portion of the TCP "Three-Way Handshake“
- When a TCP/IP stack receives a SYN packet, it responds with a SYN/ACK. which says "OK, you can connect to me, just let me make sure it's you. At this point, it is waiting for an ACK
- If the source address in the SYN packet does not exist, SYN/ACK will never be answered with an ACK, and the TCP/IP stack will wait forever for that packet

SYN Flooding (2/2)



SYN Flood: Detection

- A SYN flood attack can be detected through the use of the netstat command (**netstat –n –p TCP)**



```
tcp        1.1.1.1:80   70.56.83.204:1609      SYN_RECV
tcp        1.1.1.1:80   2.2.2.2:1723           SYN_RECV
tcp        1.1.1.1:80   209.112.192.126:4988   SYN_RECV
tcp        1.1.1.1:80   2.2.2.2:1724           SYN_RECV
tcp        1.1.1.1:80   2.2.2.2:1727           SYN_RECV
tcp        1.1.1.1:80   2.2.2.2:1733           SYN_RECV
tcp        1.1.1.1:80   24.158.121.0:3337      SYN_RECV
tcp        1.1.1.1:80   2.2.2.2:1753           SYN_RECV
tcp        1.1.1.1:80   2.2.2.2:1811           SYN_RECV
tcp        1.1.1.1:80   2.2.2.2:1821           SYN_RECV
tcp        1.1.1.1:80   2.2.2.2:1831           SYN_RECV
tcp        1.1.1.1:80   24.7.27.61:52142       SYN_RECV
tcp        1.1.1.1:80   207.118.0.58:50819     SYN_RECV
tcp        1.1.1.1:80   115.64.40.38:52865     SYN_RECV
```

SYN Flood: Prevention

- Here are some of the methods used to prevent SYN flood attacks:
  - SYN Cookies: SYN cookies ensure the server does not allocate system resources until a successful three-way handshake has been completed.
  - RST Cookies: Essentially the server responds to the client SYN frame with an incorrect SYN ACK. The client should then generate an RST packet telling the server that somethingis wrong. At this point, the server knows the client is valid and will now accept incoming connections from that client normally.
  - Micro Blocks: Micro blocks prevent SYN floods by allocating only a small space in memory for the connection record. In some cases, this memory allocation is as small as 16 bytes.
  - Stack Tweaking: This method involves changing the TCP/IP stack to prevent SYN floods. Techniques of stack tweaking include selectively dropping incoming connections or reducing the timeout when the stack will free up the memory allocated for a connection.

IP Attacks

- Smurf attack (ICMP ping→broadcast, source IP address = victim's IP address)
- UDP flood (source IP = destination IP)
- Tear Drop (packet fragmentation)