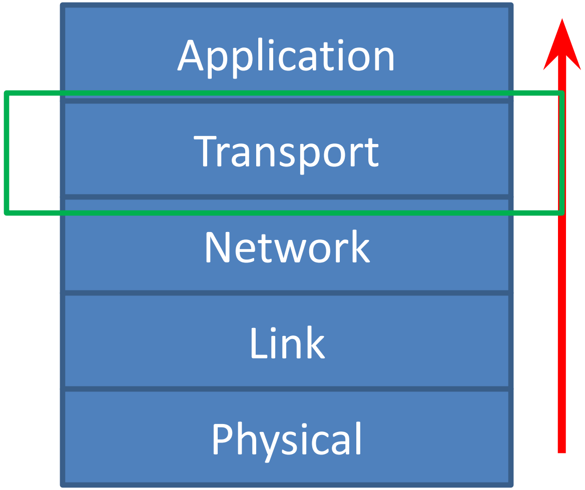
Computer and Network Security: Transport Layer Attacks and Solutions

Kameswari Chebrolu

Outline

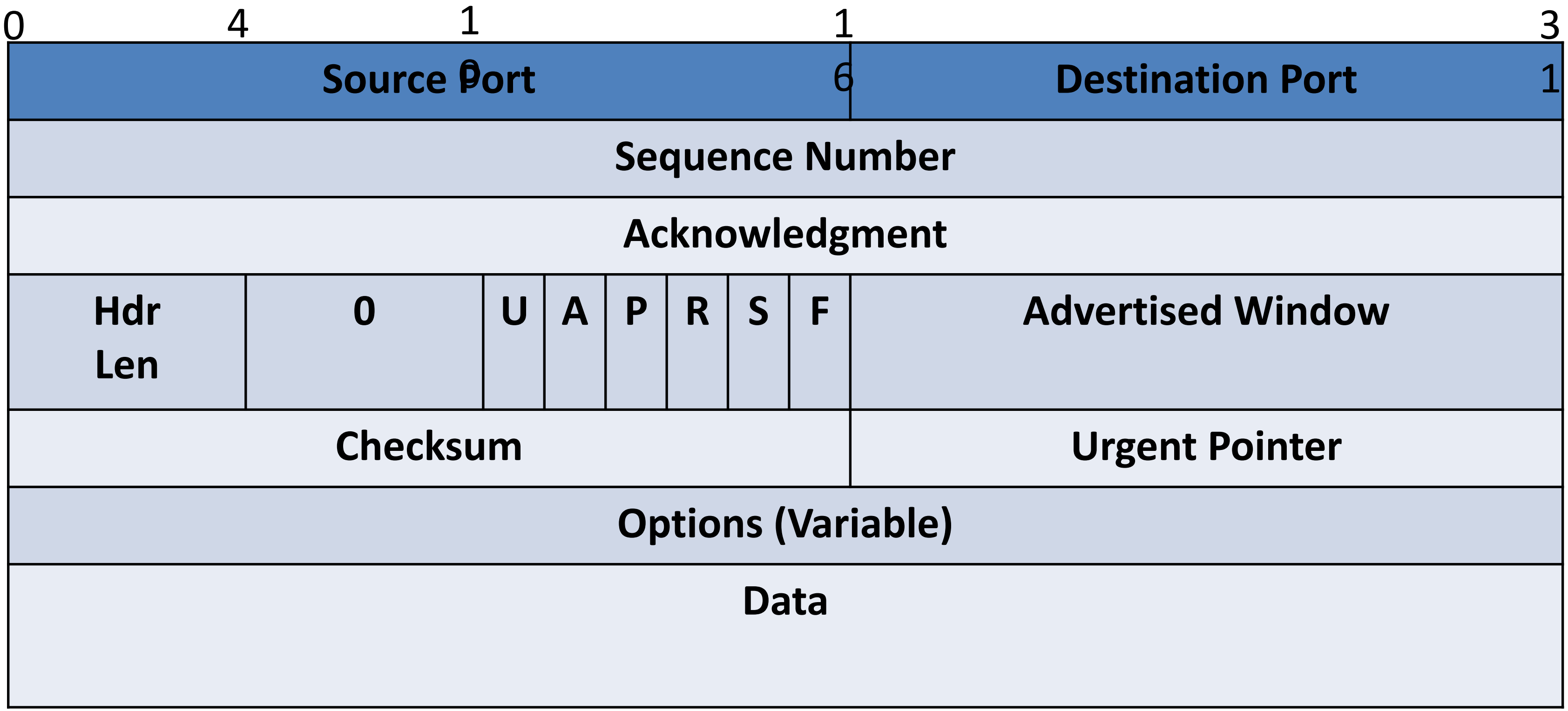
- Attacks at different layers of the protocol stack
- Solutions to the same



Transport Layer Role

- Hosts run many processes. What is the role of transport layer?
  - Process to process delivery
  - Implemented only on end-hosts
- Enhance “best-effort” network layer services to meet application expectations
- Protocols
  - UDP: Simple, provides demultiplexing
  - **TCP**: Complex, provides demultiplexing, reliability, congestion/flow control

TCP header

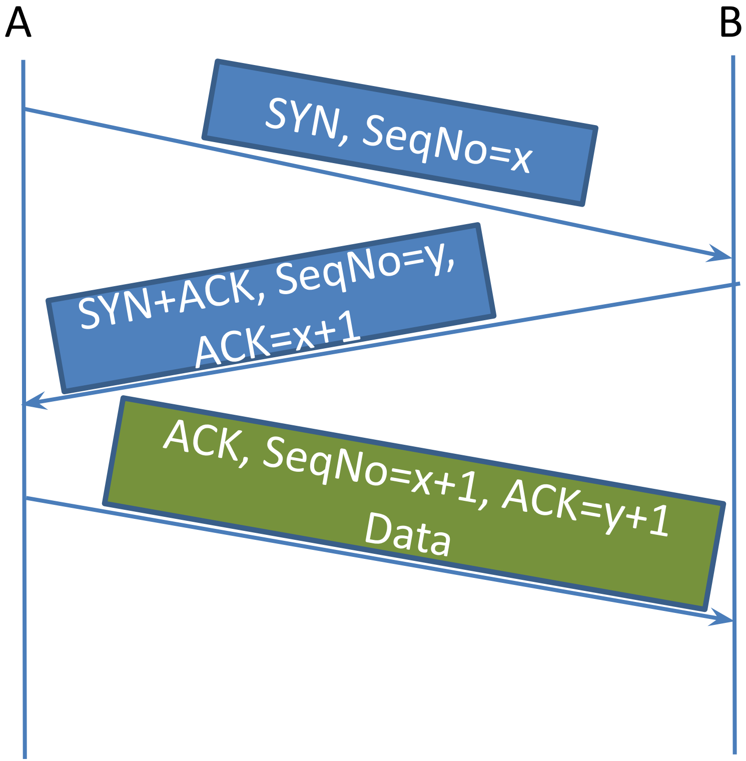


<table>
<tr><td colspan="9">Source Port</td><td>Destination Port</td></tr>
<tr><td colspan="10">Sequence Number</td></tr>
<tr><td colspan="10">Acknowledgment</td></tr>
<tr><td>Hdr
Len</td><td>0</td><td>U</td><td>A</td><td>P</td><td>R</td><td>S</td><td>F</td><td colspan="2">Advertised Window</td></tr>
<tr><td colspan="8">Checksum</td><td colspan="2">Urgent Pointer</td></tr>
<tr><td colspan="10">Options (Variable)</td></tr>
<tr><td colspan="10">Data</td></tr>
</table>

TCP connection identified by a 4 tuple: src IP, src port, dst IP, dst port

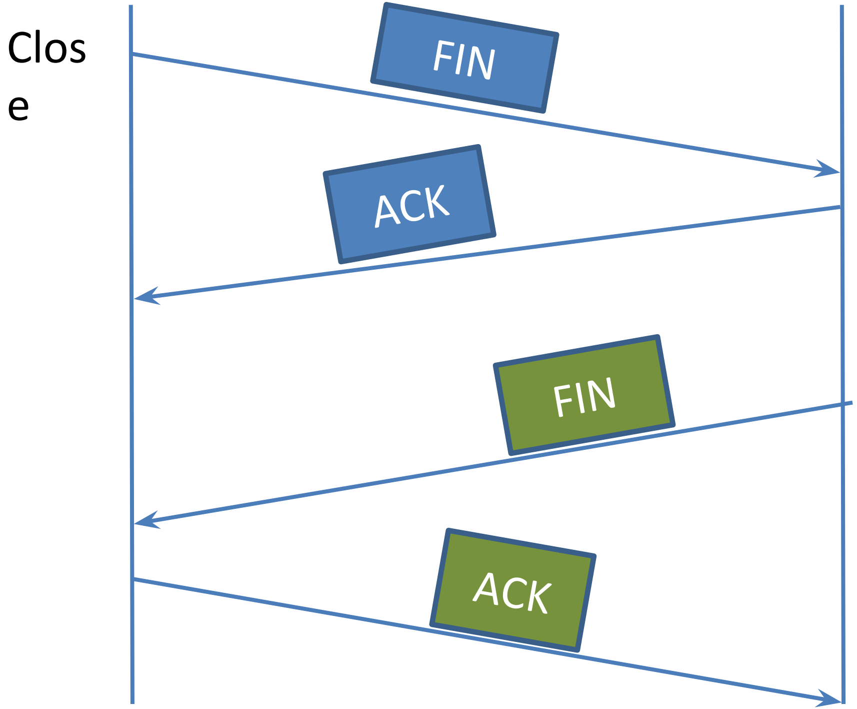
TCP: 3 Way Handshake

- Used for connection set-up
- Random initial sequence number. Why?
  - Segments from different connections can get mixed up
  - Security risk when ISN’s are predictable
    - Spoofing/hijacking (to be covered later)



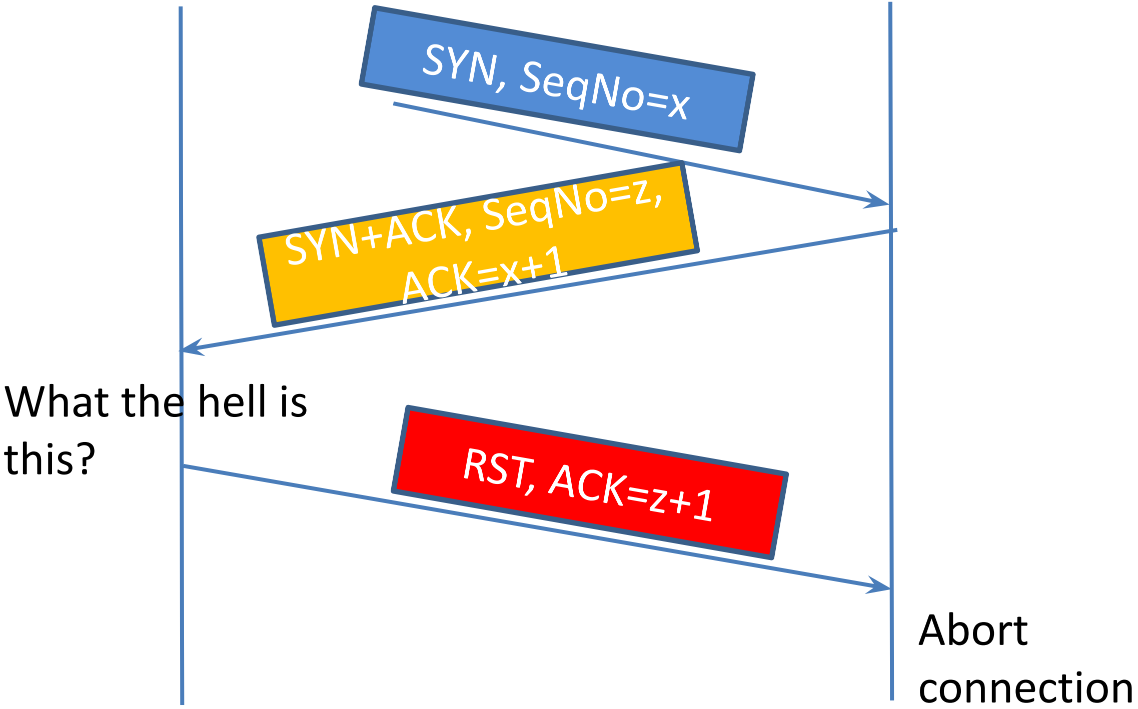
TCP: Connection Termination

- Follows simple two-way handshake
- Each side independently closes connection



Reset

- Either side can terminate connection via RST
  - Triggered by any odd behavior
  - Immediate (no ack needed)
  - Correct sequence number/port/IP is the only check



Attacks

- Focus on TCP (protocol specific attacks)
- Eavesdropping (does not make sense here; lower layer functionality)
- **Disruption**
  - **TCP SYN Flood**
  - **TCP Session Hijacking**
- **Spoofing**
  - **TCP Session Spoofing**

TCP SYN Flood Attack

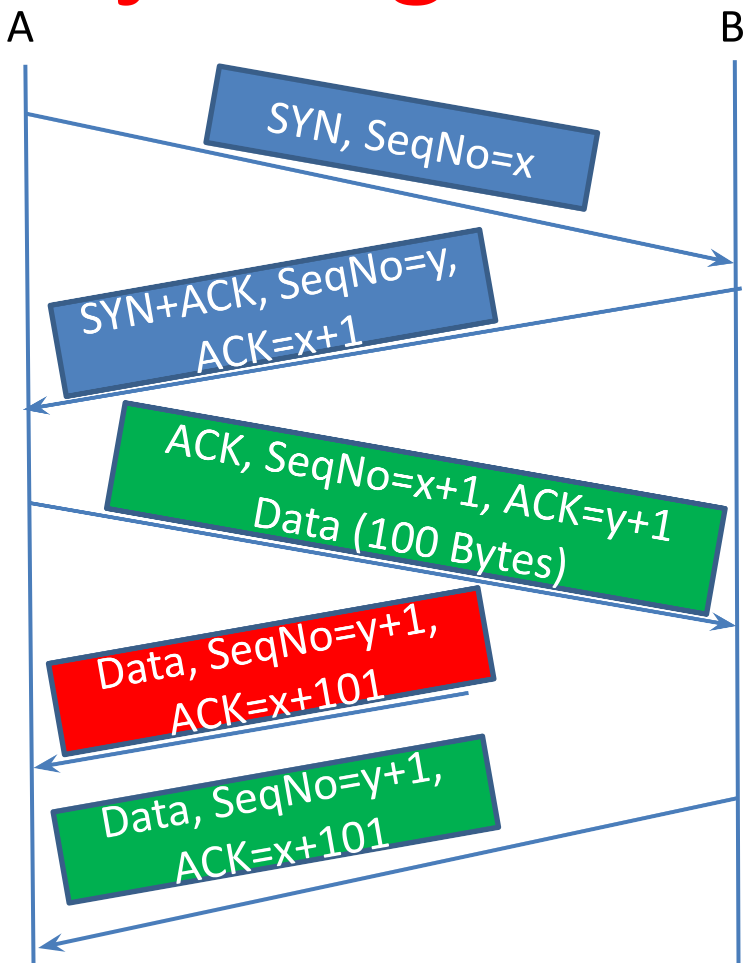
- Type of DOS attack
- Attacker sends many SYNs to target
  - When target sends SYN+ACK, does not respond with ACK
  - Connection left hanging in half-open state
  - Each new connection allocated some memory, this attack exhausts available memory at target
- Target cannot response to legitimate traffic since no memory available

(Will be covered in more detail under DOS)

TCP Session Hijacking

- Take over an already established connection
- What can one do after taking over?
  - Inject fake data that can cause damage (e.g. transfer money)
  - Close the connection (disrupt service)
- What is required to take over the connection?
  - Need to know the port, seq no information
    - Easy in wireless networks; malicious network operators (on path attacks)
    - Difficult to launch off-path attacks; but one can try to guess/infer

Injecting Data



A

B

SYN, SeqNo=x

SYN+ACK, SeqNo=y, ACK=x+1

ACK, SeqNo=x+1, ACK=y+1 Data (100 Bytes)

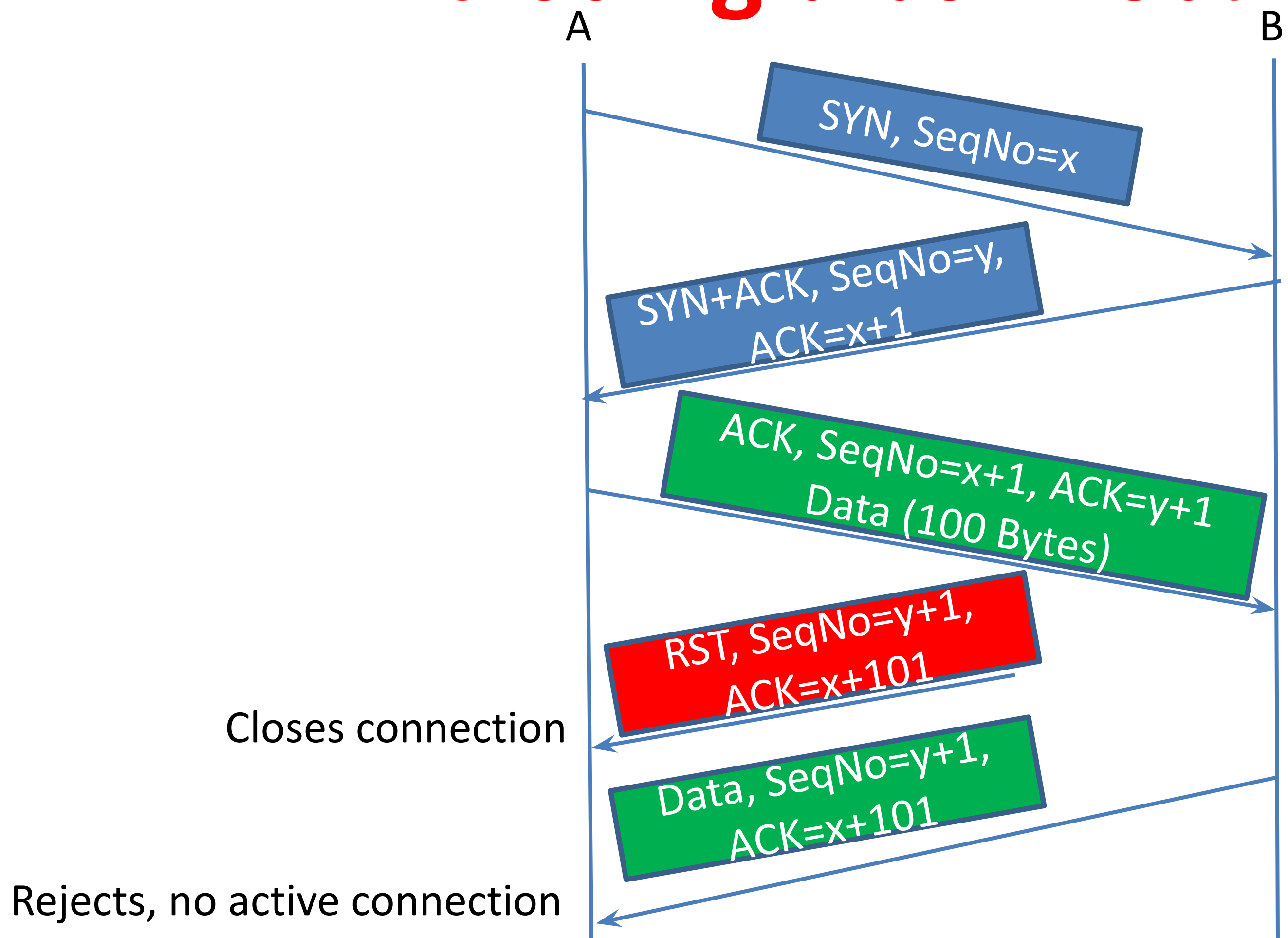
Data, SeqNo=y+1, ACK=x+101

Data, SeqNo=y+1, ACK=x+101

Dangerous Data accepted from attacker

Valid data rejected from valid end-point

Closing a connection



A

B

SYN, SeqNo=x

SYN+ACK, SeqNo=y, ACK=x+1

ACK, SeqNo=x+1, ACK=y+1 Data (100 Bytes)

RST, SeqNo=y+1, ACK=x+101

Closes connection

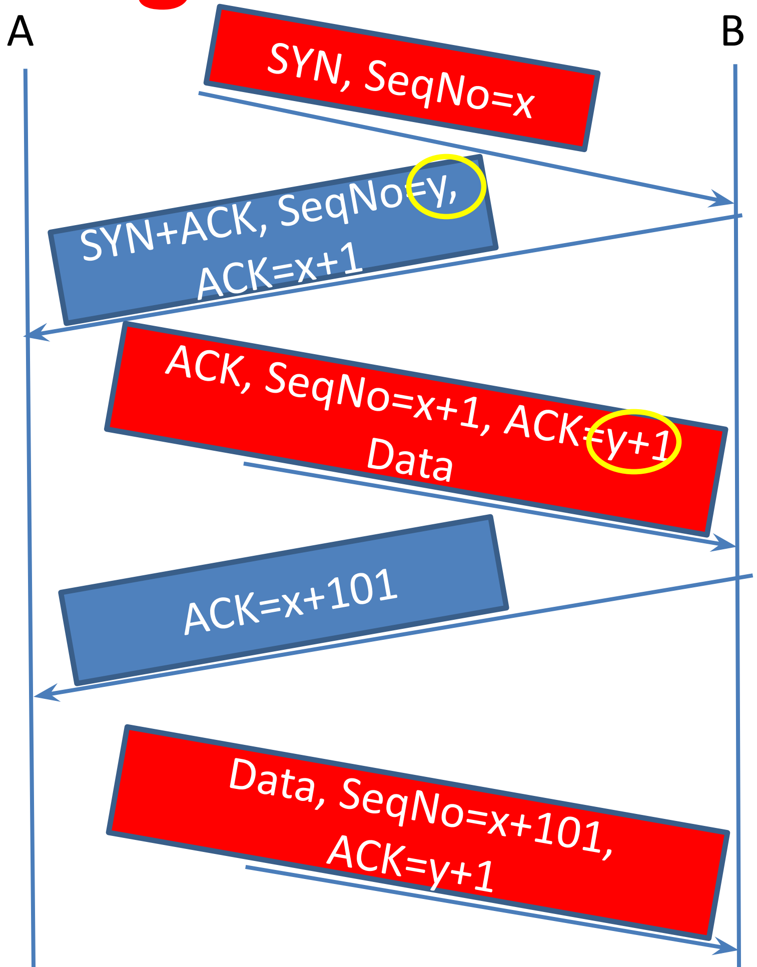
Data, SeqNo=y+1, ACK=x+101

Rejects, no active connection

TCP Session Spoofing

- Create a fake TCP connection (by taking on some one else’s IP address)
- What can one achieve?
  - Cause damage by leveraging the end point’s trust (see Mitnick attack)
- What is required to fake connection?
  - Need to know the port, **initial** seqno information
  - Bring down the machine you are imitating

Spoofing TCP Handshake



Danger of A sending RST to close the connection

Often A is brought down via DOS attack so it does not respond

How can the attacker predict y?

(see Mitnick attack)

TCP Defences Summary

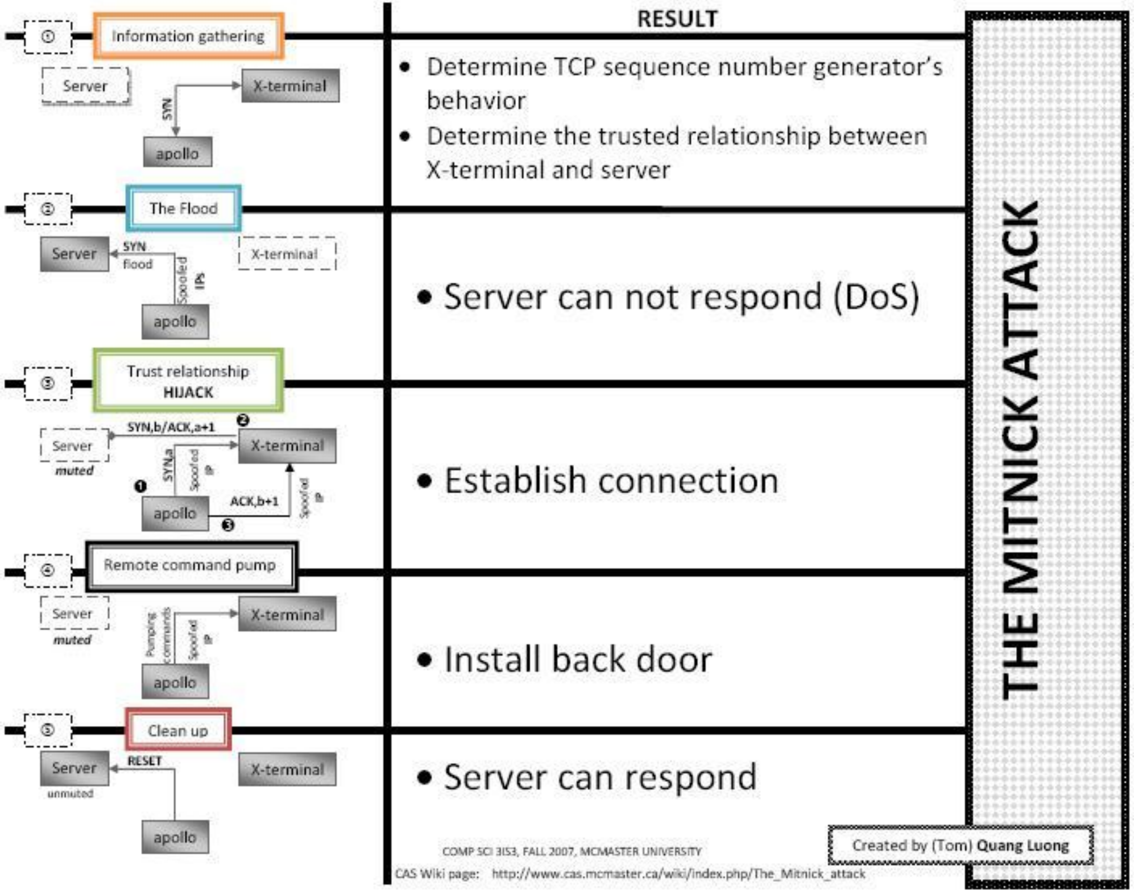
TCP SYN Flood

- Filtering, SYN Cookies, firewalls etc (to be covered later under DOS)

TCP Session hijacking/spoofing

- Choose random initial TCP sequence number
  - Handles off path attacks, but not on-path attacks
- IPsec or transport level encryption (SSL/TLS)

Mitnick Attack



RESULT

- Determine TCP sequence number generator's behavior
- Determine the trusted relationship between X-terminal and server

- Server can not respond (DoS)

- Establish connection

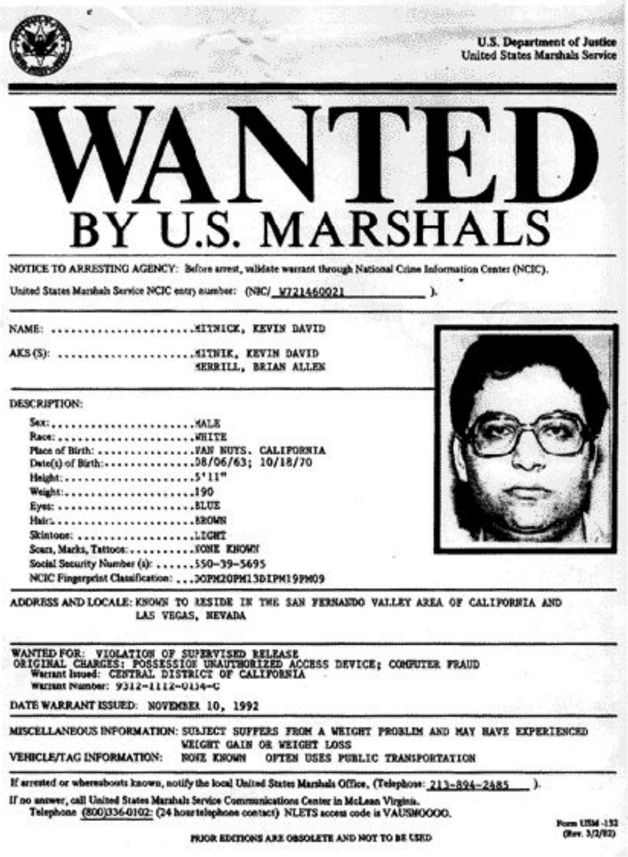
- Install back door

- Server can respond

COMP SCI 3I53, FALL 2007, MCMASTER UNIVERSITY

Created by (Tom) Quang Luong

CAS Wiki page: http://www.cas.mcmaster.ca/wiki/index.php/The_Mitnick_attack



U.S. Department of Justice
United States Marshals Service

WANTED
BY U.S. MARSHALS

NOTICE TO ARRESTING AGENCY: Before arrest, validate warrant through National Crime Information Center (NCIC).

United States Marshals Service NCIC entry number: (NIC/ W721460021).

NAME: ....MITNICK, KEVIN DAVID

AKS (S): ....MITNIK, KEVIN DAVID
MERRILL, BRIAN ALLEN

DESCRIPTION:

Sex: ....MALE
Race: ....WHITE
Place of Birth: ....VAN NUYS. CALIFORNIA
Date(s) of Birth: 08/06/63; 10/18/70
Height: ....5'11"
Weight: 190
Eyes: ....BLUE
Hair: ....BROWN
Skintone: ....LIGHT
Scars, Marks, Tattoos: ....NONE KNOWN
Social Security Number (s): 550-39-5695
NCIC Fingerprint Classification: ...DOPM20PM13DIPM19PM09

ADDRESS AND LOCALE: KNOWN TO RESIDE IN THE SAN FERNANDO VALLEY AREA OF CALIFORNIA AND LAS VEGAS, NEVADA

WANTED FOR: VIOLATION OF SUPERVISED RELEASE
ORIGINAL CHARGES: POSSESSION UNAUTHORIZED ACCESS DEVICE; COMPUTER FRAUD
Warrant Issued: CENTRAL DISTRICT OF CALIFORNIA
Warrant Number: 9312-1112-0154-G

DATE WARRANT ISSUED: NOVEMBER 10, 1992

MISCELLANEOUS INFORMATION: SUBJECT SUFFERS FROM A WEIGHT PROBLEM AND MAY HAVE EXPERIENCED WEIGHT GAIN OR WEIGHT LOSS
VEHICLE/TAG INFORMATION: NONE KNOWN OFTEN USES PUBLIC TRANSPORTATION

If arrested or whereabouts known, notify the local United States Marshals Office, (Telephone: 213-894-2485).

If no answer, call United States Marshals Service Communications Center in McLean Virginia.
Telephone (800)336-0102: (24 hour telephone contact) NLETS access code is VAUSMOOOO.

Form USM -132
(Rev. 3/2/82)

PRIOR EDITIONS ARE OBSOLETE AND NOT TO BE USED

November 1992

Information gathering

- Determine TCP sequence number
  - Send SYN to x-term; RST on receiving syn+ack. Repeat 20 times
  - Two successive TCP seq no differed by 128000
- Determine Trust relation
  - Hacked website and used command ‘finger’ and showmount to find if X-Terminal had trusted relationship with any other computers.

Other Steps

- Mute Server by TCP SYN flood attack (DOS attack)
  - Use spoofed non-routable IP addresses to send SYN requests
  - Server available memory exhausted from half-open connections
  - Server cannot respond to any more requests
- Trusted relationship hijacking
  - Establish TCP connection with x-term with source IP as Server’s (TCP session spoofing)
  - Predict x-term’s sequence number and complete 3-way handshake

Other Steps

- Remote command pump
  - Application on top of TCP is remote shell (like ssh but not secure)
  - Create a backdoor on x-term to allow any computer to connect without verification
  - Exact command:"echo + + >> /.rhosts“
- Clean up
  - Free server by send RST to cancel all SYN requests

Mitnick: Detection and Prevention

- Attack leveraged many vulnerabilities; All need addressing
- Host/network based intrusion detection and firewalls
  - for flooding; detecting attempts at information gathering; illegal access to resources
- TCP random sequence numbers (for preventing guess work)
- Using secure applications (ssh or SSL/TLS)

Summary

- Looked at TCP background
  - TCP header, connection establishment and tear down
- Disruption and Spoofing attacks
  - TCP SYN flood, TCP session hijacking, TCP session spoofing
  - Case study: Mitnick Attack
- Some solutions to the same
  - Specifically importance of random initial sequence number