TCP Protocol and Its Attacks

Lecture 07

Instructor: Dr. Cong Pu, Ph.D.

cong.pu@okstate.edu

Acknowledgment: Adapted partially from course materials from Dr. Wenliang Du at Syracuse University, Dr. Fengwei Zhang at Southern University of Science and Technology, and Dr. Steven M. Bellovin at Columbia University.

Closing TCP Connections

- When making phone call, two typical ways to disconnect
  1. two parties say goodbye to each other, then hang up (civilized)
  2. one party simply hangs up without saying goodbye (rude)

> *Rude or civilized, both methods can disconnect phone call.*

- For the "civilized" approach, when the end A of a TCP connection has no data to send, it sends out a ***FIN packet*** to the other end B.
  - ***FIN*** is one of the six code bits in the TCP header
  - after the other end B receives the ***FIN packet***
    - replies an ***ACK packet***
  - the ***A-to-B direction*** of connection is closed
    - the B-to-A direction of connection is still open

Closing TCP Connections

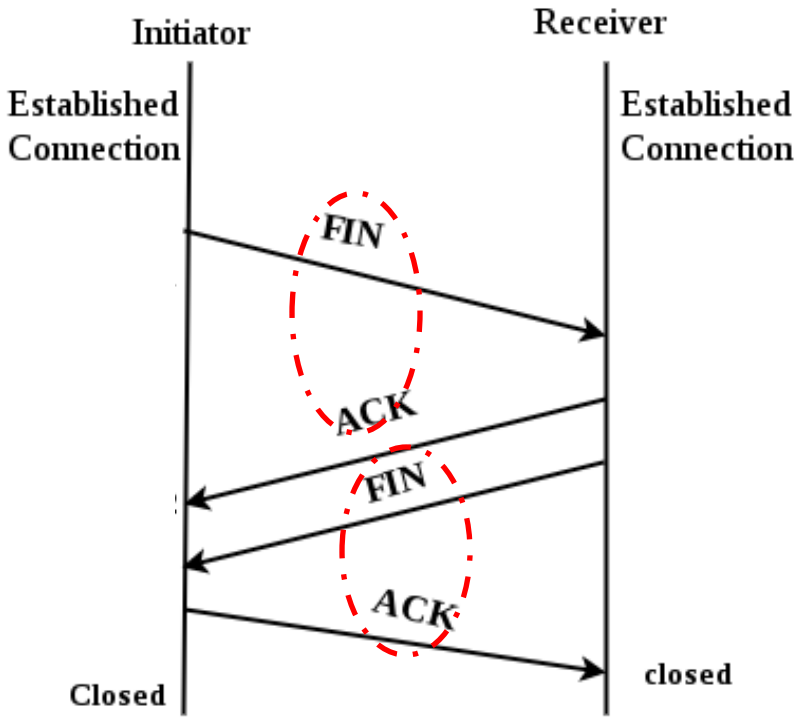
- When making phone call, two typical ways to disconnect
  1. two parties say goodbye to each other, then hang up (civilized)
  2. one party simply hangs up without saying goodbye (rude)

Rude or civilized, both methods can disconnect phone call.

- For the "civilized" approach, when the other end B of a TCP connection has no data to send, it sends out a ***FIN packet*** to the other end A.
  - ***FIN*** is one of the six code bits in the TCP header
  - after the end A receives the ***FIN packet***
    - replies an ***ACK packet***
  - the ***B-to-A direction*** of connection is closed
  - at this point, the ***entire*** TCP connection is closed

Closing TCP Connections

- When making phone call, two typical ways to disconnect
  1. two parties say goodbye to each other, then hang up (civilized)



Closing TCP Connections

- When making phone call, two typical ways to disconnect
  1. two parties say goodbye to each other, then hang up (civilized)
  2. one party simply hangs up without saying goodbye (rude)

Rude or civilized, both methods can disconnect phone call.

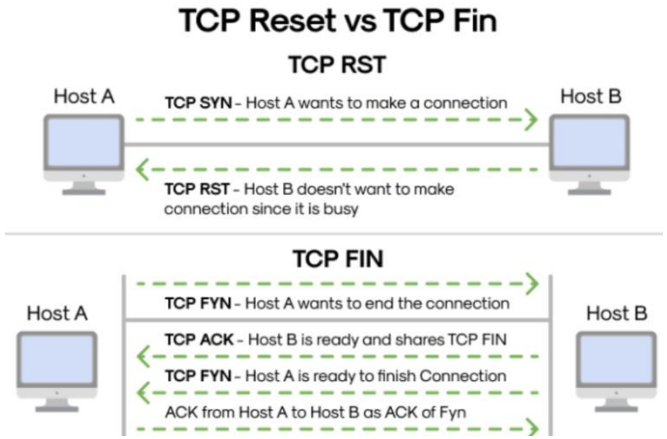
- For the "*rude*" approach, one end (e.g., A) simply sends a single ***TCP RST packet*** to the other end (e.g., B)
  - *immediately breaking the connection*
  - ***RST*** is one of the six code bits in the TCP header
    - used for *emergency situations*, e.g., no time to do ***FIN*** protocol
  - ***RST*** packets ae also sent when errors are detected
    - SYN flooding attacks; spoofed src. IP addr.

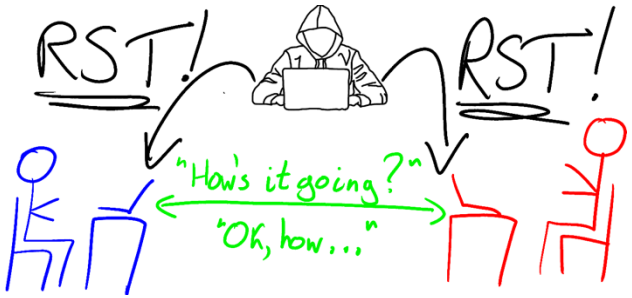
TCP Reset Attacks

- ***RST packet***: a single packet can ***CLOSE*** a TCP connection.
  - perfect candidate for attacks
- If one end can send out an ***RST packet*** to the other end to break up the connection,

What prevents an attack from sending out exactly the same packet on behalf of either end?

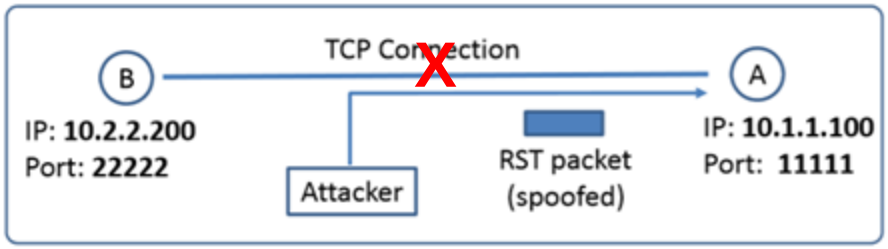
TCP Reset Attack





TCP Reset Attacks

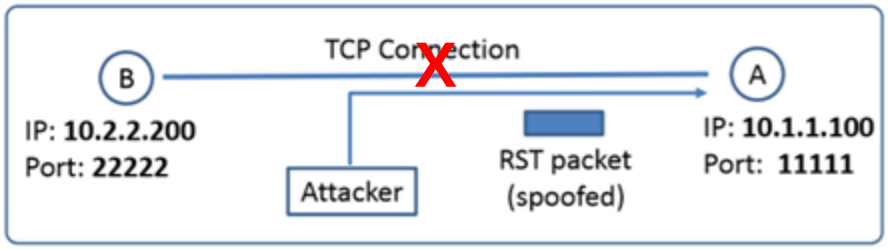
- ***Idea*** of ***TCP Reset Attacks***:
  - to break up a TCP connection between two ends, attacker just ***spoofs a TCP RST packet*** from one end to the other end



  - fill out several fields of IP and TCP headers ***correctly***
    - TCP connection *uniquely identified by four values*: ***src. IP addr.***, ***src. port #***, ***dest. IP addr.***, and ***dest. port #***
      - those four values need to be ***same*** as those used by connection

TCP Reset Attacks

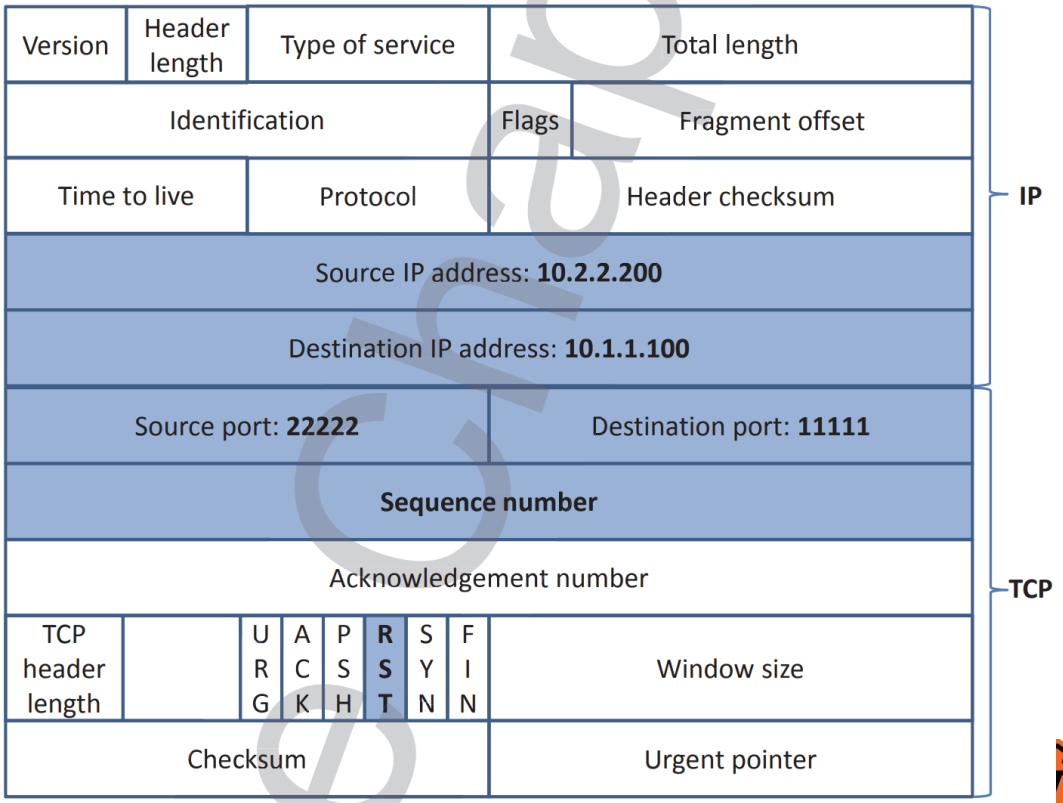
- ***Idea*** of ***TCP Reset Attacks***:
  - to break up a TCP connection between two ends, attacker just ***spoofs a TCP RST packet*** from one end to the other end



  - fill out several fields of IP and TCP headers ***correctly***
    - ensuring data is delivered accurately and in order between a sender and receiver in a network: ***sequence #***
      - otherwise discarded by the receiver
        - valid as long as sequence # is within receiver's window

TCP Reset Attacks

- *Idea* of ***TCP Reset Attacks***:
  - fill out several fields of IP and TCP headers ***correctly***
    - ***src. IP addr.***
    - ***src. port #***
    - ***dest. IP addr.***
    - ***dest. port #***
    - ***sequence #***



Launching TCP Reset Attacks: Setup

- To gain a first-hand experience on the TCP Reset attacks, we launch the attack in the VM
  - if the attacker is *not on the same network* as either the client or the server, the attack will be quite *difficult* due to *the difficulty of guessing the correct sequence #*
    - can be done in practice, but we would like to avoid
    - focusing on the *key idea* of the TCP Reset attack

TCP Reset Attacks on Video Streaming Connections

- Suppose that your roommates are watching online videos
  - most video streaming sites, e.g., YouTube and Netflix, use TCP
- *Attack goal*: break your roommates' TCP connections with the video hosting server
  - *How*: send a ***TCP RST packet*** to your roommates' machines
- *Attack challenge*: sequence number
  - in video streaming connections
    - the sequence number increases very fast due to the high data rate and continuous nature of video data
      - making manual efforts very difficult, if possible at all

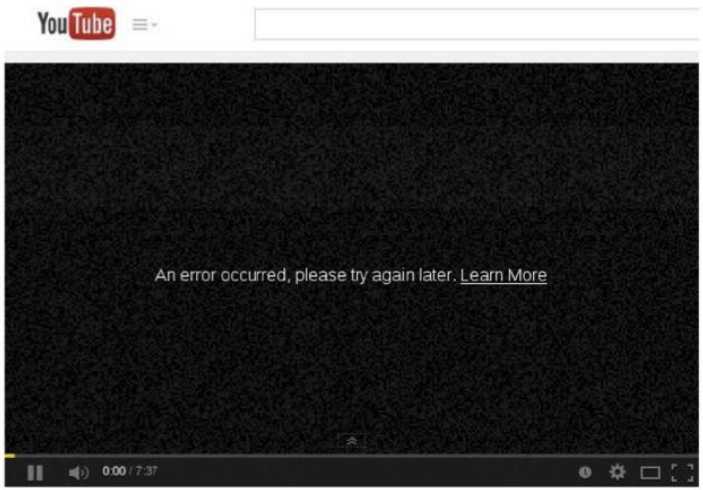
TCP Reset Attacks on Video Streaming Connections

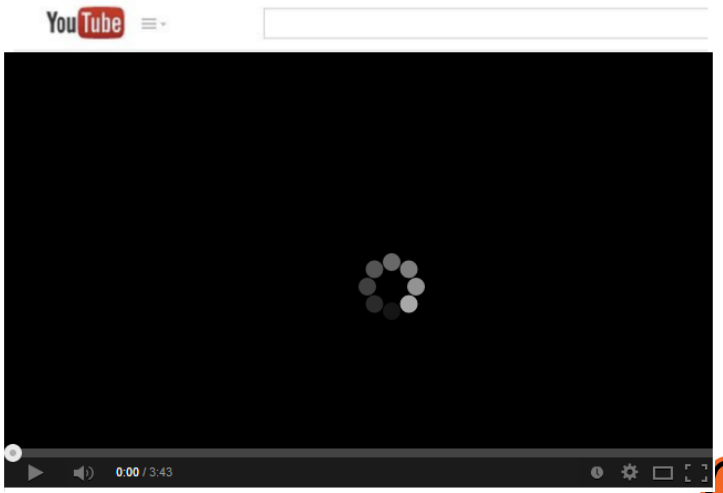
- Automate attack: using a program
  1. ***sniffs*** the video streaming packets
  2. ***get*** the sequence number and other essential parameters
  3. ***automatically sends out*** spoofed TCP RST packets
- Use Scapy to write a Python program
- Environment setup:
  - watch a YouTube video on the VM
  - run the attack program
  - Python program sends out a ***TCP RST packet*** for each packet that comes from the VM
    - the spoofed packets will go to the VM
    - resetting all of its connection, including the one with YouTube

Scapy is a powerful, interactive Python library used for network packet manipulation, packet creation, and network analysis.

TCP Reset Attacks on Video Streaming Connections

- We may not be able to see the effect immediately, even if the attack is successful
  - the video players have buffers that store a few seconds of video data ahead of what is currently being played
- Just be patient and you will see something similar to the following





TCP Reset Attacks on Video Streaming Connections

- Python code might be ***too slow to reset*** video streaming connections
  - TCP Rest attack needs to use the ***correct*** sequence number
  - when there are a lot of traffics
    - if the attack program code does not send out the spoofed TCP RST packet in time
    - the sequence number it chooses to use may has already been consumed by other packets
    - the RST packet will be discarded by the receiver

TCP Reset Attacks on Video Streaming Connections

- Another attack strategy: send out spoofed ***TCP RST packets*** using a C program
- Netwox tool 78 is such program
  - send out a TCP RST packet for each packet that comes from the VM

$ sudo netwox 78 --filter "src host xxx.xxx.xxx.xxx"

specifies a filter to match packets coming from a specific source IP address (in this case, xxx.xxx.xxx.xxx)

ip address of the target machine