

# TCP/IP Attack Lab

Copyright © 2018 Wenliang Du, Syracuse University.

The development of this document was partially funded by the National Science Foundation under Award No. 1303306 and 1718086. This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. A human-readable summary of (and not a substitute for) the license is the following: You are free to copy and redistribute the material in any medium or format. You must give appropriate credit. If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original. You may not use the material for commercial purposes.

## 1 Lab Overview

The learning objective of this lab is for students to gain first-hand experience on vulnerabilities, as well as on attacks against these vulnerabilities. Wise people learn from mistakes. In security education, we study mistakes that lead to software vulnerabilities. Studying mistakes from the past not only help students understand why systems are vulnerable, why a seemingly-benign mistake can turn into a disaster, and why many security mechanisms are needed. More importantly, it also helps students learn the common patterns of vulnerabilities, so they can avoid making similar mistakes in the future. Moreover, using vulnerabilities as case studies, students can learn the principles of secure design, secure programming, and security testing.

The vulnerabilities in the TCP/IP protocols represent a special genre of vulnerabilities in protocol designs and implementations; they provide an invaluable lesson as to why security should be designed in from the beginning, rather than being added as an afterthought. Moreover, studying these vulnerabilities help students understand the challenges of network security and why many network security measures are needed. In this lab, students need to conduct several attacks on the TCP protocol. This lab covers the following topics:

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

**Readings and related topics.** Detailed coverage of TCP attacks can be found in Chapter 13 of the SEED book, *Computer Security: A Hands-on Approach*, by Wenliang Du.

**Lab environment.** This lab has been tested on our pre-built Ubuntu 16.04 VM, which can be downloaded from the SEED website.

## 2 Lab Environment

**Network Setup.** To conduct this lab, students need to have at least 3 machines. One computer is used for attacking, the second computer is used as the victim, and the third computer is used as the observer. Students can set up 3 virtual machines on the same host computer, or they can set up 2 virtual machines, and then use the host computer as the third computer. For this lab, we put all these three machines on the same LAN, the configuration is described in Figure 1.

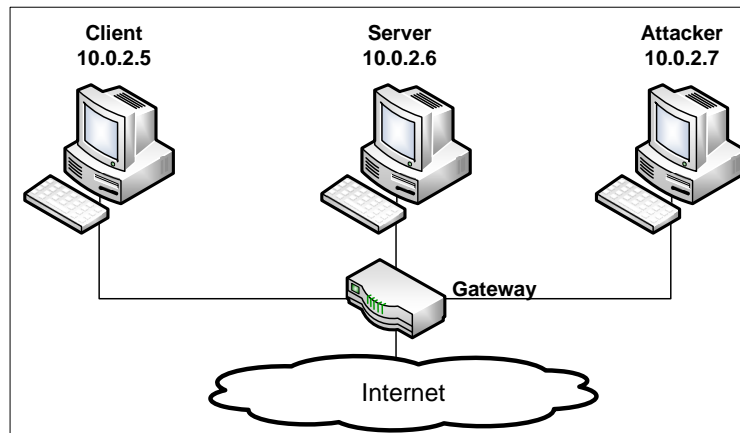


Figure 1: Environment Setup

**Netwox Tools.** We need tools to send out network packets of different types and with different contents. We can use `Netwag` to do that. However, the GUI interface of `Netwag` makes it difficult for us to automate the process. Therefore, we strongly suggest students to use its command-line version, the `Netwox` command, which is the underlying command invoked by `Netwag`.

`Netwox` consists of a suite of tools, each having a specific number. You can run a command like following (the parameters depend on which tool you are using). For some of the tool, you have to run it with the root privilege:

```
$ sudo netwox number [parameters ... ]
```

If you are not sure how to set the parameters, you can look at the manual by issuing "`netwox number --help`". You can also learn the parameter settings by running `Netwag`: for each command you execute from the graphic interface, `Netwag` actually invokes a corresponding `Netwox` command, and it displays the parameter settings. Therefore, you can simply copy and paste the displayed command.

**Scapy Tool.** Some of the tasks in this lab can also be conducted using `Scapy`, which is a powerful interactive packet manipulation program. `Scapy` is very well maintained and is widely used; while `Netwox` is not being maintained any more. There are many online tutorials on `Scapy`; we expect students to learn how to use `Scapy` from those tutorials.

### 3 Lab Tasks

In this lab, students need to conduct attacks on the TCP/IP protocols. They can use the `Netwox` tools and/or other tools in the attacks. All the attacks are performed on Linux operating systems. However, instructors can require students to also conduct the same attacks on other operating systems and compare the observations.

To simplify the “guess” of TCP sequence numbers and source port numbers, we assume that attackers are on the same physical network as the victims. Therefore, you can use sniffer tools to get that information. The following is the list of attacks that need to be implemented.

### 3.1 Task 1: SYN Flooding Attack

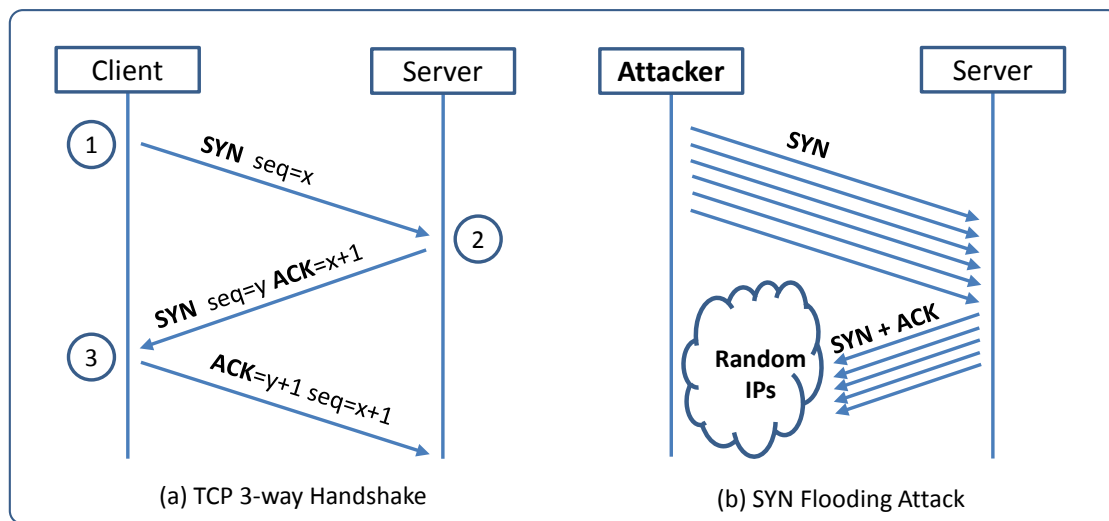


Figure 2: SYN Flooding Attack

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. Attackers either use spoofed IP address or do not continue the procedure. Through this attack, **attackers can flood the victim's queue that is used for half-opened connections, i.e. the connections that has finished SYN, SYN-ACK, but has not yet gotten a final ACK back. When this queue is full, the victim cannot take any more connection.** Figure 2 illustrates the attack.

The size of the queue has a system-wide setting. In Linux, we can check the setting using the following command:

```
$ sudo sysctl -q net.ipv4.tcp_max_syn_backlog
```

We can use command "`netstat -na`" to check the usage of the queue, i.e., the number of half-opened connection associated with a listening port. The state for such connections is `SYN-RECV`. If the 3-way handshake is finished, the state of the connections will be `ESTABLISHED`.

In this task, you need to demonstrate the SYN flooding attack. You can use the Netwox tool to conduct the attack, and then use a sniffer tool to capture the attacking packets. While the attack is going on, run the "`netstat -na`" command on the victim machine, and compare the result with that before the attack. Please also describe how you know whether the attack is successful or not.

The corresponding Netwox tool for this task is numbered 76. Here is a simple help screen for this tool. You can also type "`netwox 76 --help`" to get the help information.

Listing 1: The usage of the Netwox Tool 76

```
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 initialization type
```

**SYN Cookie Countermeasure:** If your attack seems unsuccessful, one thing that you can investigate is whether the SYN cookie mechanism is turned on. SYN cookie is a defense mechanism to counter the SYN flooding attack. The mechanism will kick in if the machine detects that it is under the SYN flooding attack. You can use the `sysctl` command to turn on/off the SYN cookie mechanism:

```
$ 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)
```

Please run your attacks with the SYN cookie mechanism on and off, and compare the results. In your report, please describe why the SYN cookie can effectively protect the machine against the SYN flooding attack. If your instructor does not cover the mechanism in the lecture, you can find out how the SYN cookie mechanism works from the Internet.

**Note on Scapy:** Although theoretically, we can use Scapy for this task, we have observed that the number of packets sent out by Scapy per second is much smaller than that by `Netwox`. This low rate makes it difficult for the attack to be successful. We were not able to succeed in SYN flooding attacks using Scapy.

### 3.2 Task 2: TCP RST Attacks on `telnet` and `ssh` Connections

The TCP RST Attack can terminate an established TCP connection between two victims. For example, 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.

In this task, you need to launch an TCP RST attack to break an existing `telnet` connection between A and B. After that, try the same attack on an `ssh` connection. Please describe your observations. To simplify the lab, we assume that the attacker and the victim are on the same LAN, i.e., the attacker can observe the TCP traffic between A and B.

**Using Netwox.** The corresponding Netwox tool for this task is numbered 78. Here is a simple help screen for this tool. You can also type "`netwox 78 --help`" to get the help information.

Listing 2: The usage of the Netwox Tool 78

```
Title: Reset every TCP packet
Usage: netwox 78 [-d device] [-f filter] [-s spoofip]
Parameters:
-d|--device device          device name {Eth0}
-f|--filter filter          pcap filter
-s|--spoofip spoofip       IP spoof initialization type {linkbraw}
```

**Using Scapy.** Please also use Scapy to conduct the TCP RST attack. A skeleton code is provided in the following (you need to replace each `@@@@` with an actual value):

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

ip  = IP(src="@@@@", dst="@@@@")
tcp = TCP(sport=@@@@, dport=@@@@, flags="@@@@", seq=@@@@, ack=@@@@)
```

```
pkt = ip/tcp
ls(pkt)
send(pkt, verbose=0)
```

### 3.3 Task 3: TCP RST Attacks on Video Streaming Applications

Let us make the TCP RST attack more interesting by experimenting it on the applications that are widely used in nowadays. We choose the video streaming application in this task. For this task, you can choose a video streaming web site that you are familiar with (we will not name any specific web site here). Most of video sharing websites establish a TCP connection with the client for streaming the video content. The attacker's goal is to disrupt the TCP session established between the victim and video streaming machine. To simplify the lab, we assume that the attacker and the victim are on the same LAN. In the following, we describe the common interaction between a user (the victim) and some video-streaming web site:

- The victim browses for a video content in the video-streaming web site, and selects one of the videos for streaming.
- Normally video contents are hosted by a different machine, where all the video contents are located. After the victim selects a video, a TCP session will be established between the victim machine and the content server for the video streaming. The victim can then view the video he/she has selected.

Your task is to disrupt the video streaming by breaking the TCP connection between the victim and the content server. You can let the victim user browse the video-streaming site from another (virtual) machine or from the same (virtual) machine as the attacker. Please be noted that, to avoid liability issues, any attacking packets should be targeted at the victim machine (which is the machine run by yourself), not at the content server machine (which does not belong to you). You only need to use `Netwox` for this task.

## 4 Lab Report

You should submit a lab report. The report should cover the following sections:

- **Design:** 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?