**CMPT 479/980:**
**Systems and Network Security**
*Spring 2020*

# Assignment 3

TCP/IP Attacks

Due date: April 9, 2020

**Instructor:** Khaled Diab

The goal of the assignment is to reproduce and launch:

(a) TCP SYN flooding attack,
(b) TCP RST attack, and
(c) TCP session hijacking attack.

## 1. Prerequisites

### 1.1 Lab VM

For this programming assignment, you will use a VM provided by SEED labs.
VM download page: https://seedsecuritylabs.org/lab_env.html
VM user manual: https://seedsecuritylabs.org/Labs_16.04/Documents/SEEDVM_VirtualBoxManual.pdf

*The VM is loaded with all required software packages.*

### 1.2 Environment

**Setup.** You need to run three VMs to perform this lab: an attacker, a victim, and an observer. For the purposes of this assignment, all three VMs should be placed on the same LAN.

**Netwox Tool.** We use this tool to send network packets with different contents. Netwox consists of a set of tools, each of which has a number. For example, you can run a Netwox tool as follows:

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

You can check the help message for each tool by running this command:

```
$ netwox <number> --help
```

**Scapy.** Some of the tasks will be conducted using scapy. Unlike Netwox, scapy is well maintained, and has extensive online resources (e.g., documentation, user guides, forums, conferences, etc.)

## 2. Tasks

### Task 1: TCP SYN Flooding Attack (30%)

TCP SYN flooding is a DoS attack which sends many SYN packets to the victim machine (e.g., a web server). To complete the attack, the attacker does not complete the three-way handshake protocol, and floods the victim with too many SYN packets.

In this task, you need to reproduce and 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 (e.g., Wireshark). While the attack is going on, run the "`netstat -na`" command on the victim machine, and compare the result with that before the attack.
The corresponding Netwox tool for this task is numbered 76. You can type "`netwox 76 --help`" to get the help information.

**SYN Cookie Countermeasure**. SYN cookie is a defense mechanism to counter the SYN flooding attack. The mechanism detects that the machine is under the SYN flooding attack. You can use the `sysctl` command to turn on/off the SYN cookie mechanism:

```
$ sudo sysctl -w net.ipv4.tcp_syncookies=0  # (turn off SYN cookie)
$ sudo sysctl -w net.ipv4.tcp_syncookies=1  # (turn on SYN cookie)
```

You need to run your attacks with the SYN cookie mechanism on and off, and compare the results.

#### Questions

(1.a) Describe how you know whether the attack is successful or not.
(1.b) Write down the Netwox command you used.
(1.c) While the countermeasure is off: show the outputs of "`netstat -na`", and a screenshot of Wireshark.
(1.d) Show the same outputs as in (1.c) while the countermeasure is on. What are your observations?
(1.e) What happens to TCP sessions (e.g., `telnet`) that were established before the attack?
(1.f) Describe why the SYN cookie can effectively protect the machine against the SYN flooding attack.

## Task 2: TCP RST Attack (30%)

TCP RST attack terminates an existing connection between two TCP endpoints. The attacker 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 a TCP RST attack to break an existing `telnet` connection between A and B. After that, try the same attack on an `ssh` connection. To simplify the lab, we assume that the attacker and the victims are on the same LAN.

**Use scapy to construct and send the TCP RST packet.**

### Questions

(2.a) What header fields did you need to modify? How did you calculate these fields?
(2.b) Show screenshots of the steps you carried out to calculate the fields.
(2.c) Show screenshots of the `telnet` and `ssh` sessions after launching the attack.
(2.d) Did your attack on an `ssh` session succeed? How is this related to the encryption done by `ssh`? Explain.
(2.e) What happens if the attackers and the victims are **not** on the same LAN? Would the attack succeed? Explain.

## Task 3: TCP Session Hijacking Attack (40%)

The goal of this attack is to inject malicious contents into an existing TCP session. If this is a `telnet` session, attackers can inject malicious commands (e.g., deleting an important file) into this session, causing the victims to execute the malicious commands.

In this task, you need to demonstrate how you can hijack a `telnet` session between two computers. For the simplicity of the task, we assume that the attacker and the victim are on the same LAN. Your goal is to get the `telnet` server to run at least the first command. Running the second command is a bonus. The commands are:

**Command 1: Deleting an existing file.**
Execute the following command on the server machine:

```
$ touch /home/seed/myfile.txt
```

The injected command is: `rm /home/seed/myfile.txt`

**Command 2: Launching a Reverse shell.**

**Use scapy to hijack the existing `telnet` session.**

### Questions

(3.a) What header fields did you need to modify? How did you calculate these fields?
(3.b) Show screenshots of the steps you carried out to calculate the fields.
(3.c) Show a screenshot of packet capture after launching the attack for Command 1.
(3.d) Discuss your observations.
(3.e) Can you use this attack to hijack an `ssh` session? Explain.
(3.f) (Bonus) Show screenshots of executing Command 2.
(3.g) (Bonus) Show a screenshot of packet capture after launching the attack for Command 2.

## 3. Submission

You are required to submit:
(1) Code: your scapy implementation for Task 2 and Task 3.
(2) Detailed report (PDF): Your report should contain the actions you performed to reproduce the attacks, observations and learned lessons, and the answers for the questions in Section 2.

The files should be compressed in a single (.zip) archive.

## 4. Policy

- Late submissions will not be graded.
- Make sure that your code is well-organized with sufficient comments.
- Any form of cheating will not be tolerated. Particularly, copying code from other students or from other sources such as the Web.
- You can discuss the assignment with other students. However, the actual coding must be your own.