# CS 5153/5053 Network Security, Spring 2023

## Project 3: TCP Attacks

## Report

### Student: Austin Tyler Conn

Contents

[Link to Source Code 3](#_Toc130914904)

[Host Environment Used 3](#_Toc130914905)

[Docker Information 3](#_Toc130914906)

[Assumptions 3](#_Toc130914907)

[Task 1 4](#_Toc130914908)

[How did you perform the attack in your VM 4](#_Toc130914909)

[Screenshots 7](#_Toc130914910)

[Was the attack successful 8](#_Toc130914911)

[Task 2 9](#_Toc130914912)

[How did you perform the attack in your VM 9](#_Toc130914913)

[Screenshots 9](#_Toc130914914)

[Was the attack successful 9](#_Toc130914915)

[Task 4 10](#_Toc130914916)

[How did you perform the attack in your VM 10](#_Toc130914917)

[Screenshots 10](#_Toc130914918)

[Was the attack successful 10](#_Toc130914919)

[Task 5 11](#_Toc130914920)

[How did you perform the attack in your VM 11](#_Toc130914921)

[Screenshots 11](#_Toc130914922)

[Was the attack successful 11](#_Toc130914923)

Link to Source Code <https://github.com/austinc3030/tcp_m11809075>

## Host Environment Used

Operating System: Ubuntu 20.04 LTS



Hardware: Google Cloud E2 Instance

Links Used for Environment Setup:

* [seed-labs/seedvm-cloud.md at master · seed-labs/seed-labs (github.com)](https://github.com/seed-labs/seed-labs/blob/master/manuals/cloud/seedvm-cloud.md)
* [seed-labs/create\_vm\_gcp.md at master · seed-labs/seed-labs (github.com)](https://github.com/seed-labs/seed-labs/blob/master/manuals/cloud/create_vm_gcp.md)

## Docker Information

Text

Description automatically generated

Text

Description automatically generated

## Assumptions

1. Mapping between PDF document and docker containers provided:
   1. Client (10.0.2.5) = user1-10.9.0.6 (10.9.0.6)
   2. Server (10.0.2.6) = victim-10.9.0.5 (10.9.0.5)
   3. Attacker (10.0.2.7) = seed-attacker (10.9.0.1)

## Task 1

### How did you perform the attack in your VM

1. Write code for scapy.

Text

Description automatically generated

1. Check the size of net.ipv4.tcp\_max\_syn\_backlog on the victim/server.

Graphical user interface, text, application

Description automatically generated

1. Check the connections on the victim/server.

Text

Description automatically generated

1. Initiate a telnet session from user1/client to the victim/server.

Text

Description automatically generated

1. Check connections on the victim/server to see the new telnet connection.

Text

Description automatically generated

1. Disable SYN cookies on the victim/server per the assignment instructions (Note: the SEED Lab Docker Image for the victim/server already has SYN cookies disabled.)

Text

Description automatically generated

1. From the attacker, initiate a SYN attack using code from step 1.A screenshot of a computer

   Description automatically generated with medium confidence
2. Attempt to initiate a new telnet session from user1/client to the victim/server.

Text

Description automatically generated

1. Check netstat on the victim/server to see the active connections.

A picture containing background pattern

Description automatically generated

*Note: Full output of netstat -nat above, truncated output below for readability.*

A screenshot of a computer

Description automatically generated with medium confidence

### Screenshots

See screenshots in “How did you perform the attack in your VM”

### Was the attack successful

Yes, the attack was successful. I did find that running only 1 instance of *task1.py* seemed to have intermittent effects in that sometimes the telnet session would establish a connection and allow me to log in. I believe this may be due to the single instance of *task1.py* potentially not creating enough SYN packets fast enough to overwhelm the victim/server and as the victim/server frees a resource, the telnet session from the user1/client is allowed to establish. This makes sense considering other DOS attacks I am familiar with where it was not a single IP or machine causing the DOS but rather a botnet or network of many computers causing the DOS.

Further evidence to support that the attack was successful is the output of *netstat -nat* where it shows many ‘foreign’ IP addresses that are random and implausible in this lab network as our network is in the 10.9.0.0 address space.

*task1.py* could further be improved by allowing arguments to be passed to the script such as target IP, target port rather than having them hardcoded in the script. Also, making use of a parallel process such as python’s “threads” to spawn multiple loops, each sending out SYN packets. This would eliminate the need to run multiple instances from the command line.

## Task 2

### How did you perform the attack in your VM

### Screenshots

### Was the attack successful

## Task 4

### How did you perform the attack in your VM

### Screenshots

### Was the attack successful

## Task 5

### How did you perform the attack in your VM

### Screenshots

### Was the attack successful