Performing a Denial-of-Service Attack from the WAN

ETHICAL HACKING & LAB ASSIGNMENT 3

Student Info
Name: SRIJA PABBA
Student ID: 00866719
Email:
spabb6@unh.newhaven.edu

Table of Contents

Executive Summary

Highlights

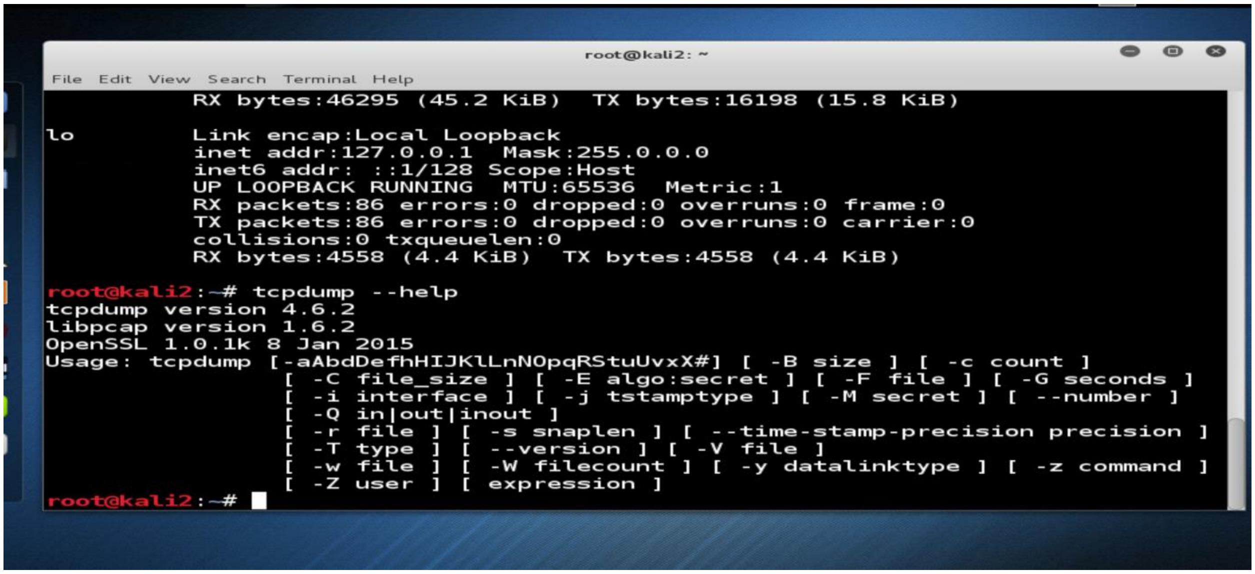
In this lab, I will explore denial-of-service (DoS) attacks to understand how different network protocols can be flooded to overwhelm target systems and disrupt services. I will conduct TCP, UDP, and HTTP flood attacks, starting by configuring the network interface on a Kali sniffer using ifconfig and capturing packets with tcpdump. Using the Low Orbit Ion Cannon (LOIC) on a Windows machine, I will generate extensive traffic and analyze it with capinfos to measure the data volume and impact. This will give me insight into how quickly such attacks can exhaust network resources and compromise service availability.

Objectives

My objectives in this lab are to understand DoS attack mechanics and their impact on network resources by executing TCP, UDP, and HTTP floods. I will set up the network, capture packets, and analyze traffic using tools like ifconfig, tcpdump, and capinfos on Kali Linux to observe how each attack overwhelms services. By simulating attack conditions with LOIC from an external network, I will learn to identify the unique traffic patterns for each protocol. Observing the high traffic volumes generated will help me understand the strain DoS attacks place on networks, and I'll gain practical experience with sniffer tools for real-world network analysis.

Lab Description Details

Below command to see all of the available options for tcpdump

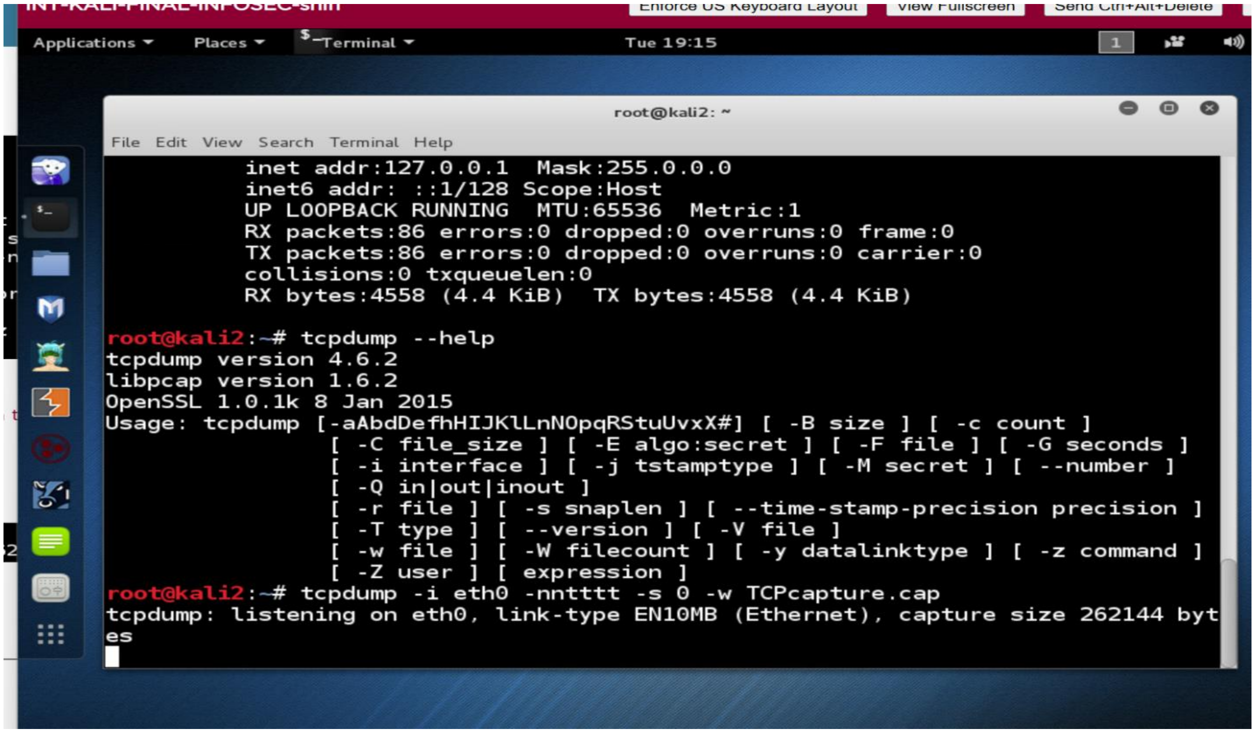


```
          RX bytes:46295 (45.2 KiB)  TX bytes:16198 (15.8 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:86 errors:0 dropped:0 overruns:0 frame:0
          TX packets:86 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:4558 (4.4 KiB)  TX bytes:4558 (4.4 KiB)

root@kali2:~# tcpdump --help
tcpdump version 4.6.2
libpcap version 1.6.2
OpenSSL 1.0.1k 8 Jan 2015
Usage: tcpdump [-aAbdDefhHIJKlLnNOpqRStuUvxX#] [ -B size ] [ -c count ]
                [ -C file_size ] [ -E algo:secret ] [ -F file ] [ -G seconds ]
                [ -i interface ] [ -j tstamptype ] [ -M secret ] [ --number ]
                [ -Q in|out|inout ]
                [ -r file ] [ -s snaplen ] [ --time-stamp-precision precision ]
                [ -T type ] [ --version ] [ -V file ]
                [ -w file ] [ -W filecount ] [ -y datalinktype ] [ -z command ]
                [ -Z user ] [ expression ]
root@kali2:~# 
```

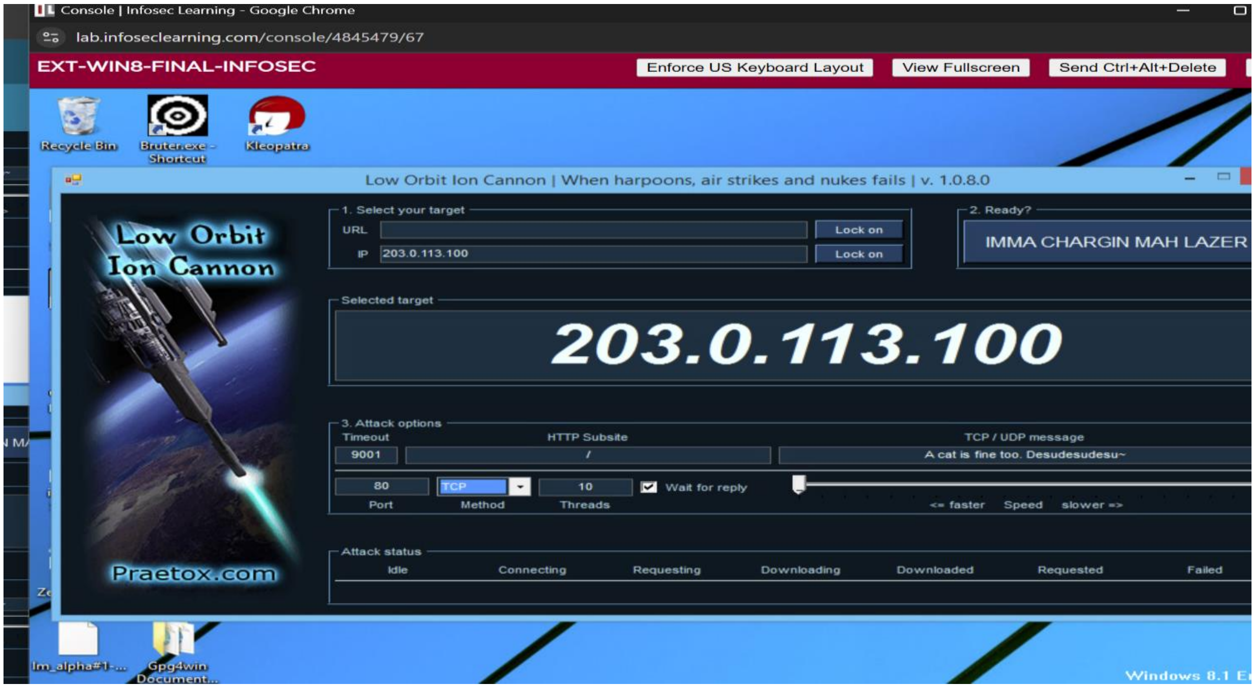
To start tcpdump sniffing on the eth0 interface


INT-KALI-FINAL-INFOSEC-sniff
Enforce US Keyboard Layout
View Fullscreen
Send Ctrl+Alt+Delete
Applications
Places
Terminal
Tue 19:15
root@kali2: ~
File Edit View Search Terminal Help

```
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:86 errors:0 dropped:0 overruns:0 frame:0
          TX packets:86 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:4558 (4.4 KiB)  TX bytes:4558 (4.4 KiB)

root@kali2:~# tcpdump --help
tcpdump version 4.6.2
libpcap version 1.6.2
OpenSSL 1.0.1k 8 Jan 2015
Usage: tcpdump [-aAbdDefhHIJKlLnNOpqRStuUvxX#] [ -B size ] [ -c count ]
                [ -C file_size ] [ -E algo:secret ] [ -F file ] [ -G seconds ]
                [ -i interface ] [ -j tstamptype ] [ -M secret ] [ --number ]
                [ -Q in|out|inout ]
                [ -r file ] [ -s snaplen ] [ --time-stamp-precision precision ]
                [ -T type ] [ --version ] [ -V file ]
                [ -w file ] [ -W filecount ] [ -y datalinktype ] [ -z command ]
                [ -Z user ] [ expression ]
root@kali2:~# tcpdump -i eth0 -nntttt -s 0 -w TCPcapture.cap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 byt
es
```

In Low Orbit Ion Cannon we locked 203.0.113.100 as our targeted IP address. And selected TCP as protocol, by clicking “IMMA CHARGIN MAH LAZER” and after 30 secs stopping the flooding button will give the packets captured and packets received by filters.



Low Orbit Ion Cannon | When harpoons, air strikes and nukes fails | v. 1.0.8.0

1. Select your target
URL
IP 203.0.113.100

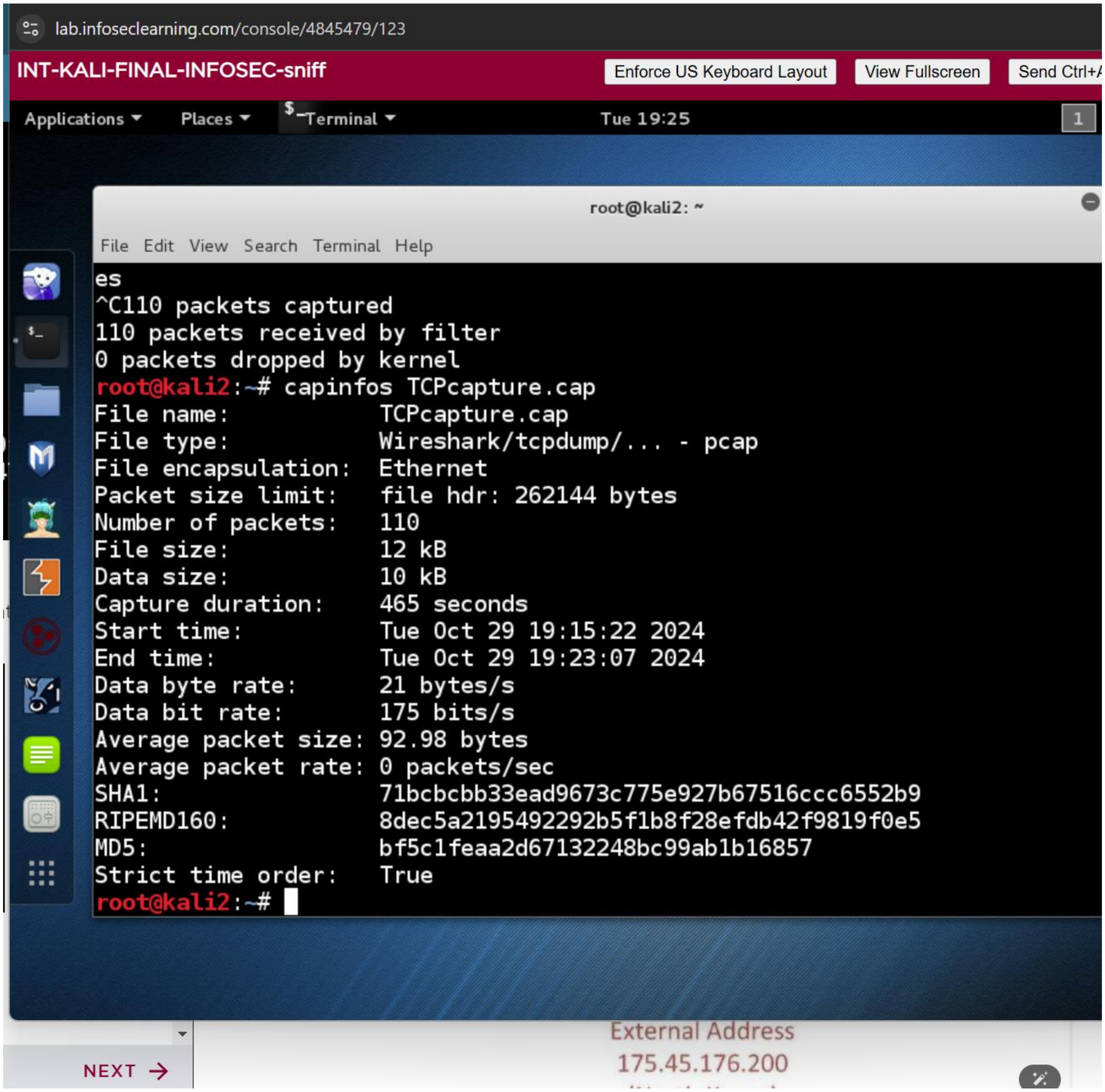
2. Ready?

Selected target
203.0.113.100

3. Attack options
Timeout 9001
HTTP Subsite /
TCP / UDP message A cat is fine too. Desudesudesu~
Port 80
Method TCP
Threads 10
☑ Wait for reply
<= faster Speed slower =>

Attack status
Idle Connecting Requesting Downloading Downloaded Requested Failed

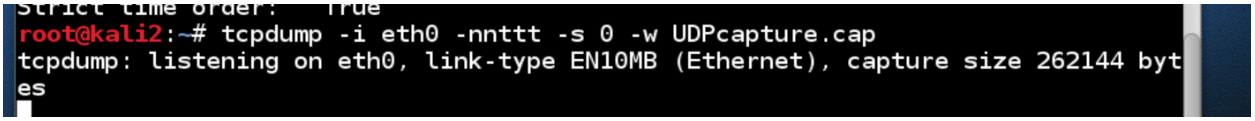
To view the total of packet in the TCPcapture file


lab.infoseclearning.com/console/4845479/123
INT-KALI-FINAL-INFOSEC-sniff
Enforce US Keyboard Layout
View Fullscreen
Send Ctrl+A
Applications
Places
Terminal
Tue 19:25
1
root@kali2: ~
File Edit View Search Terminal Help
NEXT →
External Address
175.45.176.200

```
es
^C110 packets captured
110 packets received by filter
0 packets dropped by kernel
root@kali2:~# capinfos TCPcapture.cap
File name:           TCPcapture.cap
File type:           Wireshark/tcpdump/... - pcap
File encapsulation:  Ethernet
Packet size limit:   file hdr: 262144 bytes
Number of packets:   110
File size:           12 kB
Data size:           10 kB
Capture duration:    465 seconds
Start time:          Tue Oct 29 19:15:22 2024
End time:            Tue Oct 29 19:23:07 2024
Data byte rate:      21 bytes/s
Data bit rate:       175 bits/s
Average packet size: 92.98 bytes
Average packet rate: 0 packets/sec
SHA1:                71bcbcbb33ead9673c775e927b67516ccc6552b9
RIPEMD160:           8dec5a2195492292b5f1b8f28efdb42f9819f0e5
MD5:                 bf5c1feaa2d67132248bc99ab1b16857
Strict time order:   True
root@kali2:~# 
```

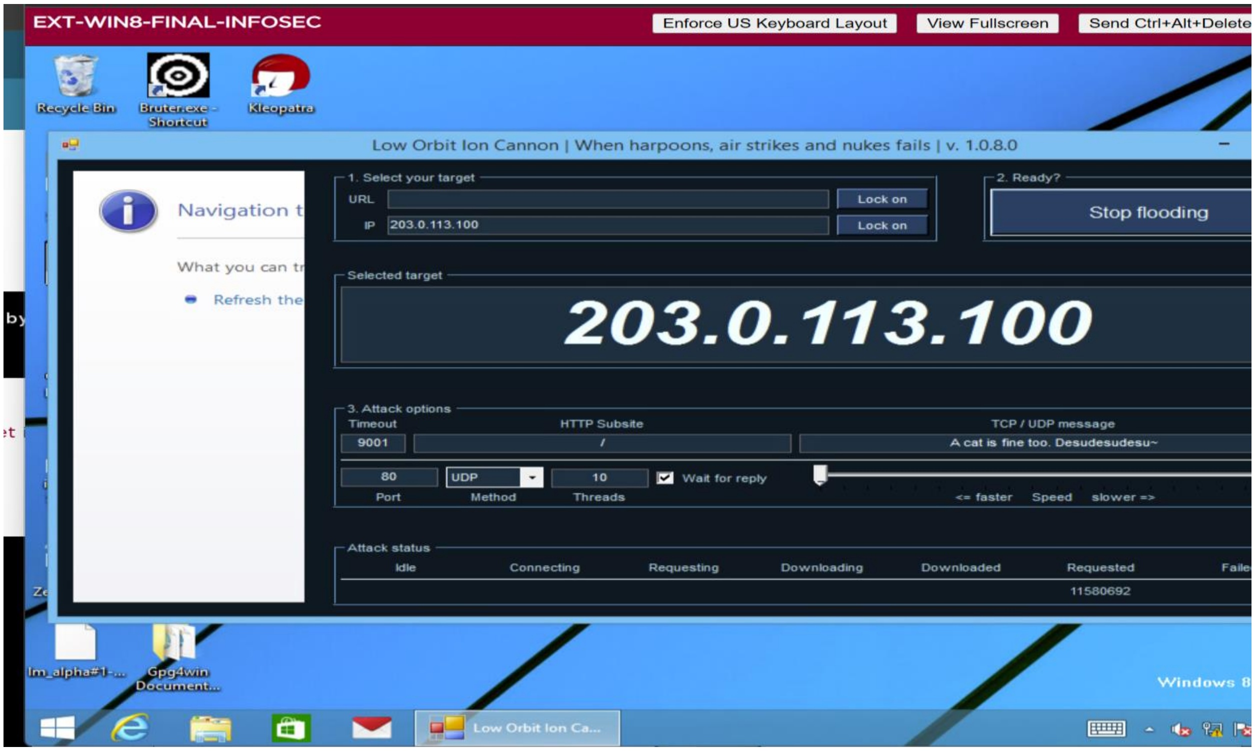
The picture shows the total number of packets captured in the number of packets data

To start tcpdump sniffing on the eth0 interafce for UDP



```
Strict time order:   True
root@kali2:~# tcpdump -i eth0 -nnttt -s 0 -w UDPcapture.cap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 byt
es
```

In Low Orbit Ion Cannon we locked 203.0.113.100 as our targeted IP address. And selected UDP as protocol, by clicking “IMMA CHARGIN MAH LAZER” and after 30 secs stopping the flooding button will give the packets captured and packets received by filters.



Low Orbit Ion Cannon | When harpoons, air strikes and nukes fails | v. 1.0.8.0

1. Select your target
URL
IP 203.0.113.100

2. Ready?

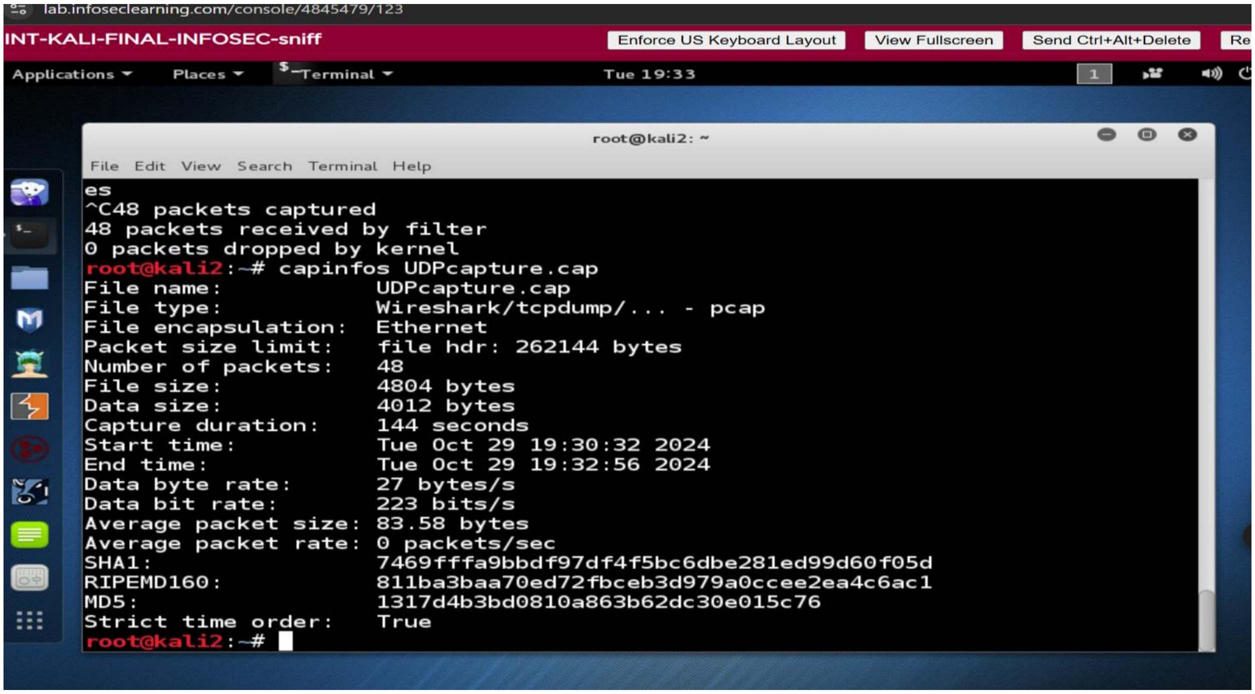
Selected target
203.0.113.100

3. Attack options

Timeout: 9001
HTTP Subsite: /
TCP / UDP message: A cat is fine too. Desudesudesu~
Port: 80
Method: UDP
Threads: 10
☑ Wait for reply
<= faster Speed slower =>

Attack status

Idle | Connecting | Requesting | Downloading | Downloaded | Requested | Faile
Requested: 11580692

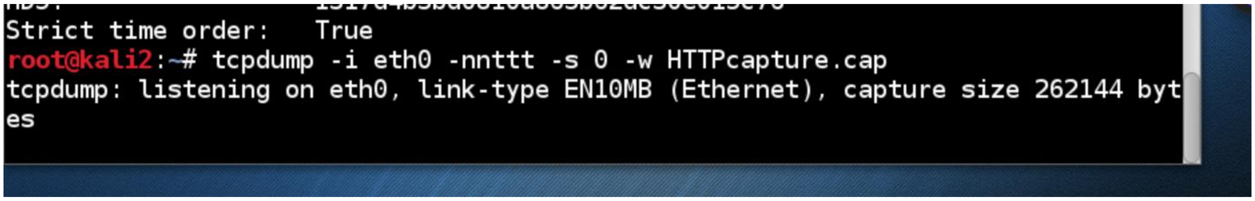


lab.infoseclearning.com/console/4845479/123
INT-KALI-FINAL-INFOSEC-sniff
Enforce US Keyboard Layout
View Fullscreen
Send Ctrl+Alt+Delete
Re
Applications
Places
Terminal
Tue 19:33
root@kali2: ~
File Edit View Search Terminal Help

```
es
^C48 packets captured
48 packets received by filter
0 packets dropped by kernel
root@kali2:~# capinfos UDPcapture.cap
File name:           UDPcapture.cap
File type:           Wireshark/tcpdump/... - pcap
File encapsulation:  Ethernet
Packet size limit:   file hdr: 262144 bytes
Number of packets:   48
File size:           4804 bytes
Data size:           4012 bytes
Capture duration:    144 seconds
Start time:          Tue Oct 29 19:30:32 2024
End time:            Tue Oct 29 19:32:56 2024
Data byte rate:      27 bytes/s
Data bit rate:       223 bits/s
Average packet size: 83.58 bytes
Average packet rate: 0 packets/sec
SHA1:                7469fffa9bbdf97df4f5bc6dbe281ed99d60f05d
RIPEMD160:           811ba3baa70ed72fbceb3d979a0ccee2ea4c6ac1
MD5:                 1317d4b3bd0810a863b62dc30e015c76
Strict time order:   True
root@kali2:~# 
```

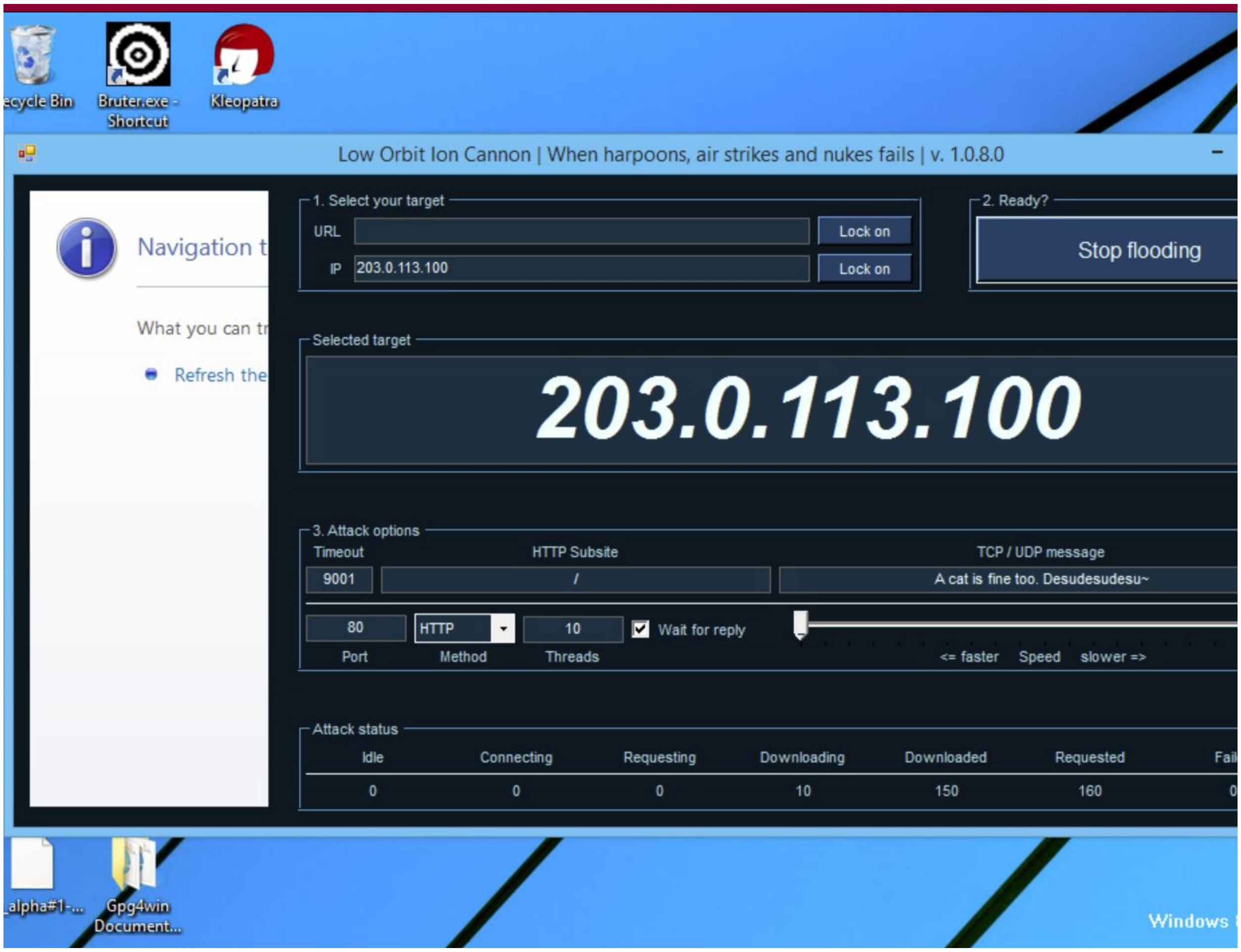
The picture shows the total number of packets captured in the number of packets data

To start the tcpdump sniffing on the eth0 interface for HTTPS



```
Strict time order:   True
root@kali2:~# tcpdump -i eth0 -nnttt -s 0 -w HTTPcapture.cap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 byt
es
```

In Low Orbit Ion Cannon we locked 203.0.113.100 as our targeted IP address. And selected HTTP as protocol, by clicking "IMMA CHARGIN MAH LAZER" and after 30 secs stopping the flooding button will give the packets captured and packets received by filters.



Low Orbit Ion Cannon | When harpoons, air strikes and nukes fails | v. 1.0.8.0

1. Select your target
URL [ ] Lock on
IP 203.0.113.100 Lock on

2. Ready?
Stop flooding

Selected target
203.0.113.100

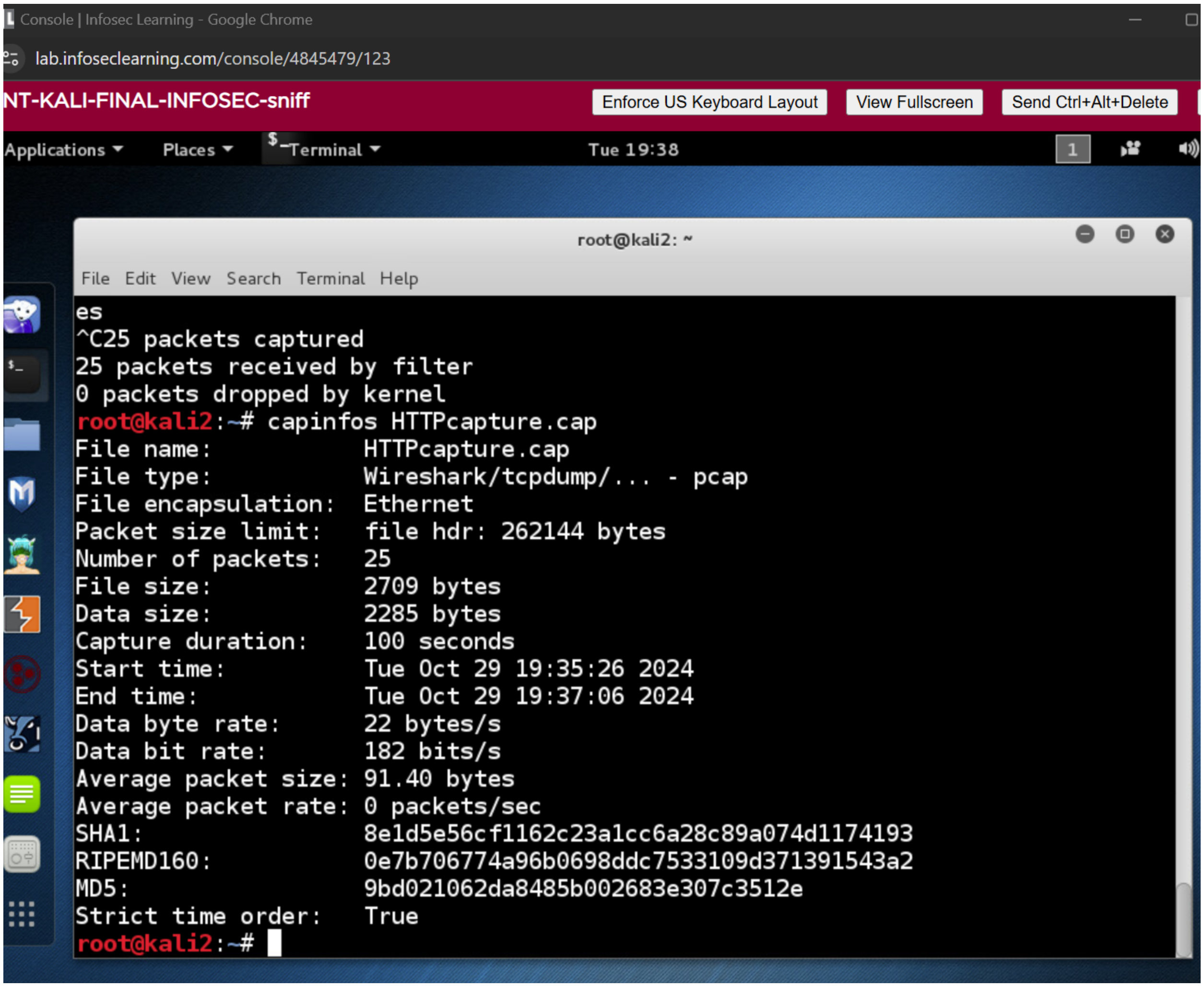
3. Attack options
Timeout: 9001
HTTP Subsite: /
TCP / UDP message: A cat is fine too. Desudesudesu~
Port: 80
Method: HTTP
Threads: 10
☑ Wait for reply
<= faster Speed slower =>

Attack status

Idle	Connecting	Requesting	Downloading	Downloaded	Requested	Fail
0	0	0	10	150	160	0

Image shows the LOW ORBIT ION CANNON interface

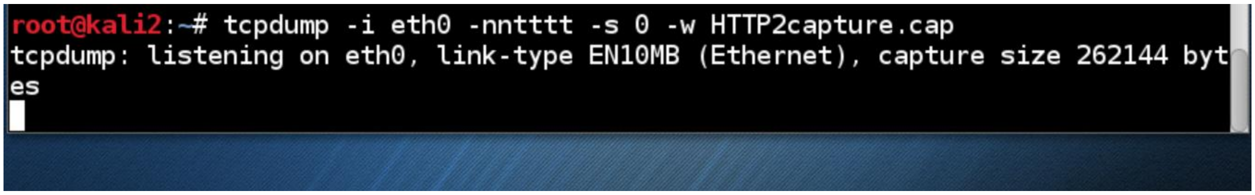
To view the total number of packets in HTTPcapture.cap file


Console | Infosec Learning - Google Chrome
lab.infoseclearning.com/console/4845479/123
NT-KALI-FINAL-INFOSEC-sniff
Enforce US Keyboard Layout
View Fullscreen
Send Ctrl+Alt+Delete
Applications
Places
Terminal
Tue 19:38
root@kali2: ~
File Edit View Search Terminal Help

```
es
^C25 packets captured
25 packets received by filter
0 packets dropped by kernel
root@kali2:~# capinfos HTTPcapture.cap
File name:           HTTPcapture.cap
File type:           Wireshark/tcpdump/... - pcap
File encapsulation:  Ethernet
Packet size limit:   file hdr: 262144 bytes
Number of packets:   25
File size:           2709 bytes
Data size:           2285 bytes
Capture duration:    100 seconds
Start time:          Tue Oct 29 19:35:26 2024
End time:            Tue Oct 29 19:37:06 2024
Data byte rate:      22 bytes/s
Data bit rate:       182 bits/s
Average packet size: 91.40 bytes
Average packet rate: 0 packets/sec
SHA1:                8e1d5e56cf1162c23a1cc6a28c89a074d1174193
RIPEMD160:           0e7b706774a96b0698ddc7533109d371391543a2
MD5:                 9bd021062da8485b002683e307c3512e
Strict time order:   True
root@kali2:~# 
```

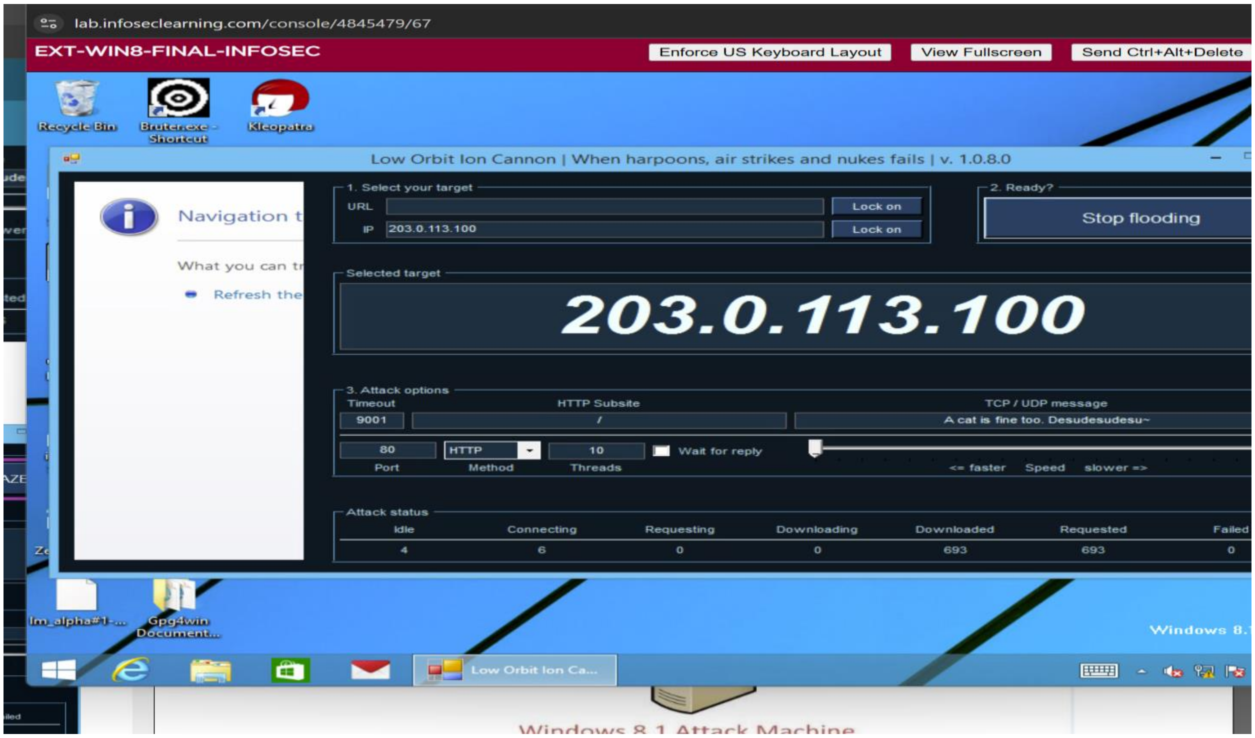
Image shows the total number of packets captured in the number of packets data

To start tcpdump sniffing on the eth0 interface for HTTP



```
root@kali2:~# tcpdump -i eth0 -nntttt -s 0 -w HTTP2capture.cap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 byt
es
```

In Low Orbit Ion Cannon we locked 203.0.113.100 as our targeted IP address. And selected HTTP as protocol, and we unchecked the wait for reply button for this. By clicking “IMMA CHARGIN MAH LAZER” and after 30 secs stopping the flooding button will give the packets captured and packets received by filters.



lab.infoseclearning.com/console/4845479/67

EXT-WIN8-FINAL-INFOSEC

Low Orbit Ion Cannon | When harpoons, air strikes and nukes fails | v. 1.0.8.0

1. Select your target
URL: Lock on
IP: 203.0.113.100 Lock on

2. Ready? Stop flooding

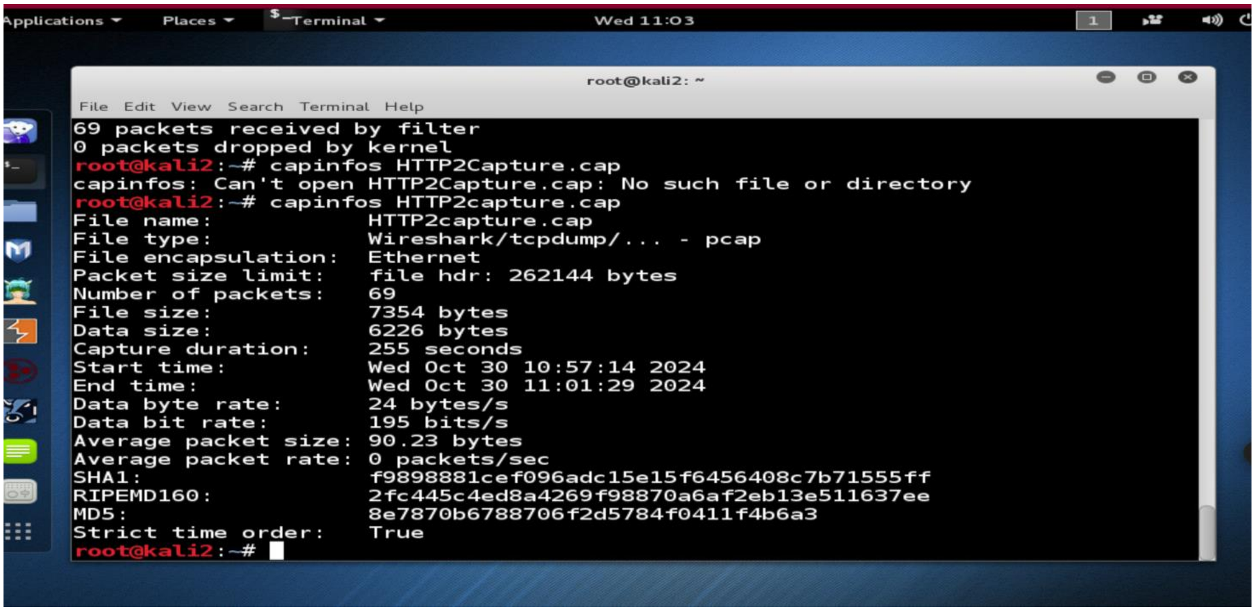
Selected target: 203.0.113.100

3. Attack options
Timeout: 9001
HTTP Subsite: /
TCP / UDP message: A cat is fine too. Desudesudesu~
Port: 80
Method: HTTP
Threads: 10
□ Wait for reply
<= faster Speed slower =>

Attack status

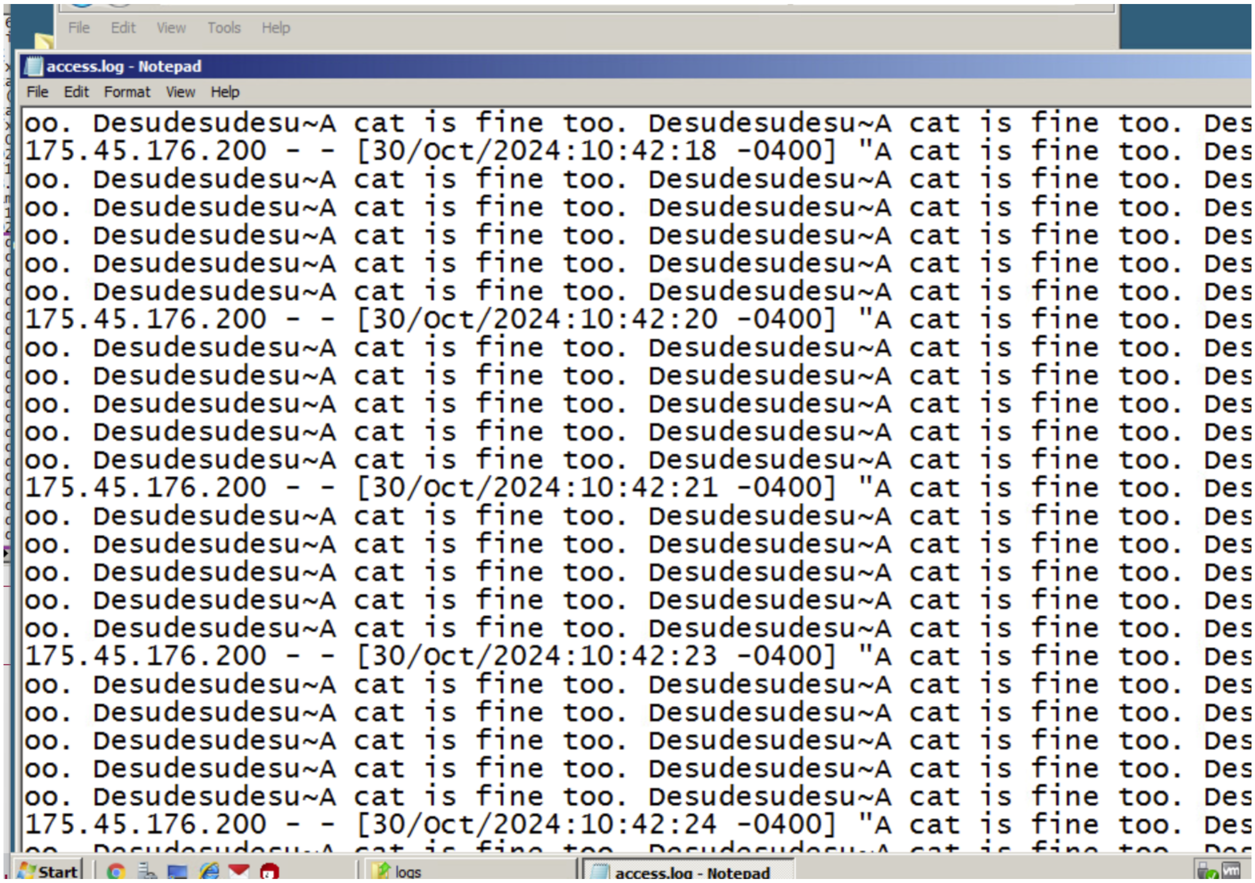
Idle	Connecting	Requesting	Downloading	Downloaded	Requested	Failed
4	6	0	0	693	693	0

This shows the total number of packets in the HTTP2capture.cap file



```
69 packets received by filter
0 packets dropped by kernel
root@kali2:~# capinfos HTTP2Capture.cap
capinfos: Can't open HTTP2Capture.cap: No such file or directory
root@kali2:~# capinfos HTTP2capture.cap
File name:           HTTP2capture.cap
File type:           Wireshark/tcpdump/... - pcap
File encapsulation:  Ethernet
Packet size limit:   file hdr: 262144 bytes
Number of packets:   69
File size:           7354 bytes
Data size:           6226 bytes
Capture duration:    255 seconds
Start time:          Wed Oct 30 10:57:14 2024
End time:            Wed Oct 30 11:01:29 2024
Data byte rate:      24 bytes/s
Data bit rate:       195 bits/s
Average packet size: 90.23 bytes
Average packet rate: 0 packets/sec
SHA1:                f9898881cef096adc15e15f6456408c7b71555ff
RIPEMD160:           2fc445c4ed8a4269f98870a6af2eb13e511637ee
MD5:                 8e7870b6788706f2d5784f0411f4b6a3
Strict time order:   True
root@kali2:~#
```

This picture shows the number of packets captured in the number of packet data

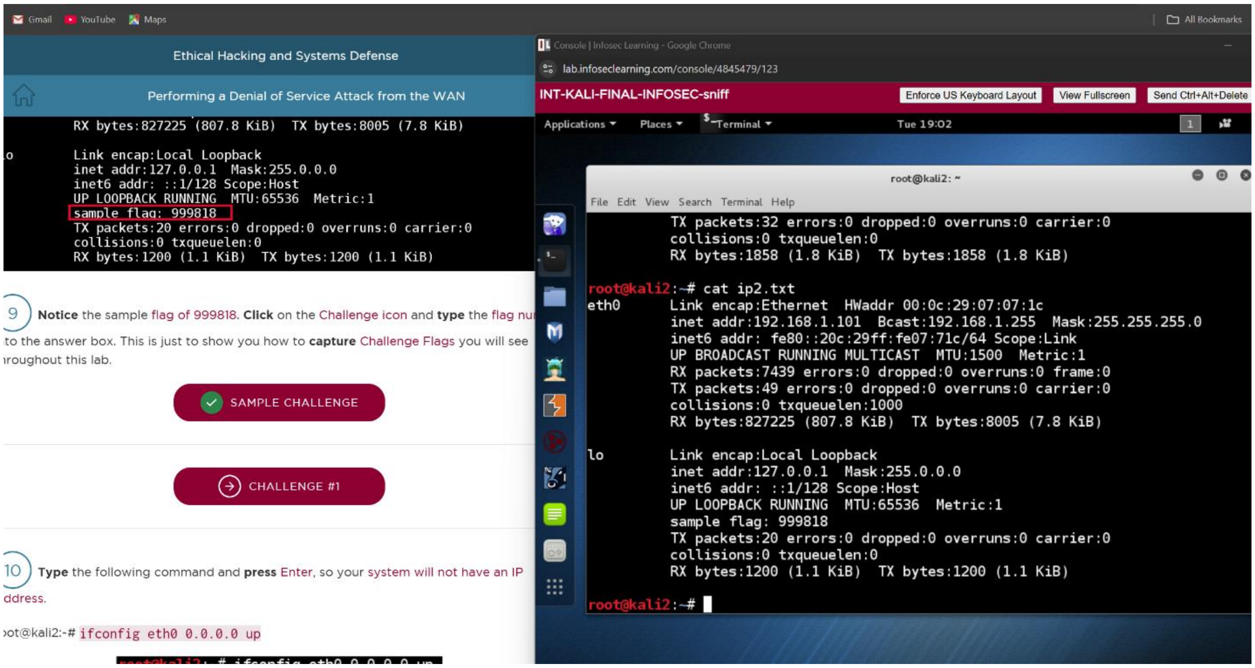


```
oo. Desudesudesu~A cat is fine too. Desudesudesu~A cat is fine too. Des
175.45.176.200 - - [30/Oct/2024:10:42:18 -0400] "A cat is fine too. Des
oo. Desudesudesu~A cat is fine too. Desudesudesu~A cat is fine too. Des
oo. Desudesudesu~A cat is fine too. Desudesudesu~A cat is fine too. Des
oo. Desudesudesu~A cat is fine too. Desudesudesu~A cat is fine too. Des
oo. Desudesudesu~A cat is fine too. Desudesudesu~A cat is fine too. Des
oo. Desudesudesu~A cat is fine too. Desudesudesu~A cat is fine too. Des
175.45.176.200 - - [30/Oct/2024:10:42:20 -0400] "A cat is fine too. Des
oo. Desudesudesu~A cat is fine too. Desudesudesu~A cat is fine too. Des
oo. Desudesudesu~A cat is fine too. Desudesudesu~A cat is fine too. Des
oo. Desudesudesu~A cat is fine too. Desudesudesu~A cat is fine too. Des
oo. Desudesudesu~A cat is fine too. Desudesudesu~A cat is fine too. Des
oo. Desudesudesu~A cat is fine too. Desudesudesu~A cat is fine too. Des
175.45.176.200 - - [30/Oct/2024:10:42:21 -0400] "A cat is fine too. Des
oo. Desudesudesu~A cat is fine too. Desudesudesu~A cat is fine too. Des
oo. Desudesudesu~A cat is fine too. Desudesudesu~A cat is fine too. Des
oo. Desudesudesu~A cat is fine too. Desudesudesu~A cat is fine too. Des
oo. Desudesudesu~A cat is fine too. Desudesudesu~A cat is fine too. Des
oo. Desudesudesu~A cat is fine too. Desudesudesu~A cat is fine too. Des
175.45.176.200 - - [30/Oct/2024:10:42:23 -0400] "A cat is fine too. Des
oo. Desudesudesu~A cat is fine too. Desudesudesu~A cat is fine too. Des
oo. Desudesudesu~A cat is fine too. Desudesudesu~A cat is fine too. Des
oo. Desudesudesu~A cat is fine too. Desudesudesu~A cat is fine too. Des
oo. Desudesudesu~A cat is fine too. Desudesudesu~A cat is fine too. Des
oo. Desudesudesu~A cat is fine too. Desudesudesu~A cat is fine too. Des
175.45.176.200 - - [30/Oct/2024:10:42:24 -0400] "A cat is fine too. Des
oo. Desudesudesu~A cat is fine too. Desudesudesu~A cat is fine too. Des
```

This picture shows the entries that state "A cat is fine too"

Supporting Evidence

These are the challenges to complete during the lab


Gmail
YouTube
Maps
All Bookmarks
Console | Infosec Learning - Google Chrome
lab.infoseclearning.com/console/4845479/123
INT-KALI-FINAL-INFOSEC-sniff
Enforce US Keyboard Layout
View Fullscreen
Send Ctrl+Alt+Delete
Applications
Places
Terminal
Tue 19:02
root@kali2: ~
File Edit View Search Terminal Help

Ethical Hacking and Systems Defense

Performing a Denial of Service Attack from the WAN

```
          RX bytes:827225 (807.8 KiB)  TX bytes:8005 (7.8 KiB)

o         Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          sample flag: 999818
          TX packets:20 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1200 (1.1 KiB)  TX bytes:1200 (1.1 KiB)
```

9 **Notice** the sample flag of 999818. **Click** on the Challenge icon and **type** the flag nu
to the answer box. This is just to show you how to **capture** Challenge Flags you will see
roughout this lab.

SAMPLE CHALLENGE

CHALLENGE #1

10 **Type** the following command and **press** Enter, so your system will not have an IP
ddress.

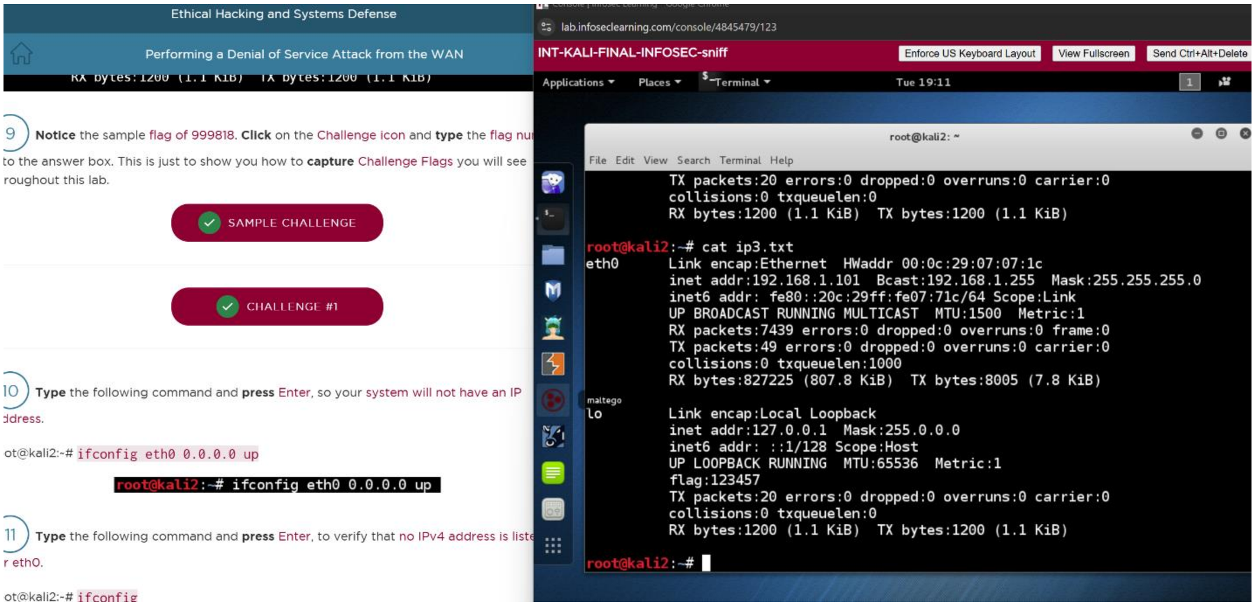
```
ot@kali2:-# ifconfig eth0 0.0.0.0 up
```

```
          TX packets:32 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1858 (1.8 KiB)  TX bytes:1858 (1.8 KiB)

root@kali2:~# cat ip2.txt
eth0      Link encap:Ethernet  HWaddr 00:0c:29:07:07:1c
          inet addr:192.168.1.101  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe07:71c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7439 errors:0 dropped:0 overruns:0 frame:0
          TX packets:49 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:827225 (807.8 KiB)  TX bytes:8005 (7.8 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          sample flag: 999818
          TX packets:20 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1200 (1.1 KiB)  TX bytes:1200 (1.1 KiB)

root@kali2:~# 
```


lab.infoseclearning.com/console/4845479/123
INT-KALI-FINAL-INFOSEC-sniff
Enforce US Keyboard Layout
View Fullscreen
Send Ctrl+Alt+Delete
Applications
Places
Terminal
Tue 19:11
root@kali2: ~
File Edit View Search Terminal Help
maltego

Ethical Hacking and Systems Defense

Performing a Denial of Service Attack from the WAN

```
          RX bytes:1200 (1.1 KiB)  TX bytes:1200 (1.1 KiB)
```

9 **Notice** the sample flag of 999818. **Click** on the Challenge icon and **type** the flag nu
to the answer box. This is just to show you how to **capture** Challenge Flags you will see
roughout this lab.

SAMPLE CHALLENGE

CHALLENGE #1

10 **Type** the following command and **press** Enter, so your system will not have an IP
ddress.

```
ot@kali2:~# ifconfig eth0 0.0.0.0 up
```

```
root@kali2:~# ifconfig eth0 0.0.0.0 up
```

11 **Type** the following command and **press** Enter, to verify that no IPv4 address is liste
r eth0.

```
ot@kali2:~# ifconfig
```

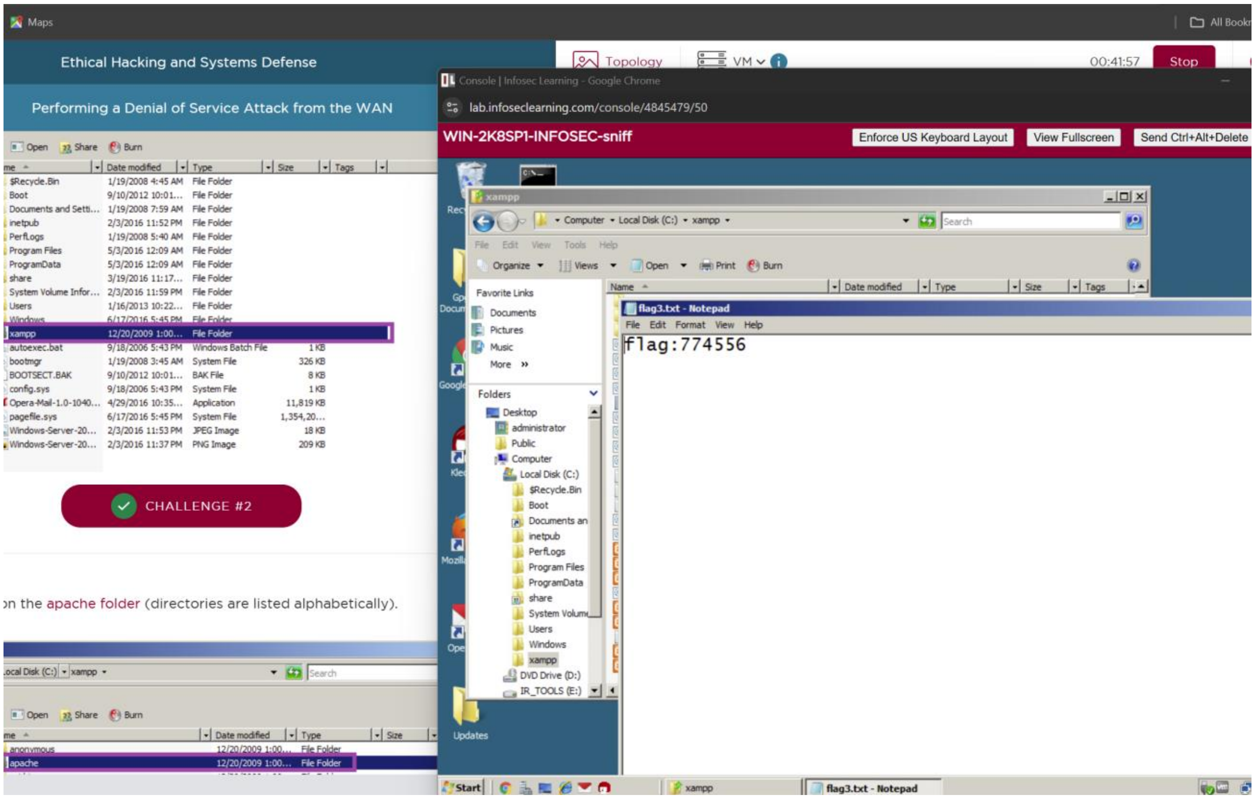
```
          TX packets:20 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1200 (1.1 KiB)  TX bytes:1200 (1.1 KiB)

root@kali2:~# cat ip3.txt
eth0      Link encap:Ethernet  HWaddr 00:0c:29:07:07:1c
          inet addr:192.168.1.101  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe07:71c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7439 errors:0 dropped:0 overruns:0 frame:0
          TX packets:49 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:827225 (807.8 KiB)  TX bytes:8005 (7.8 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          flag:123457
          TX packets:20 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1200 (1.1 KiB)  TX bytes:1200 (1.1 KiB)

root@kali2:~# 
```

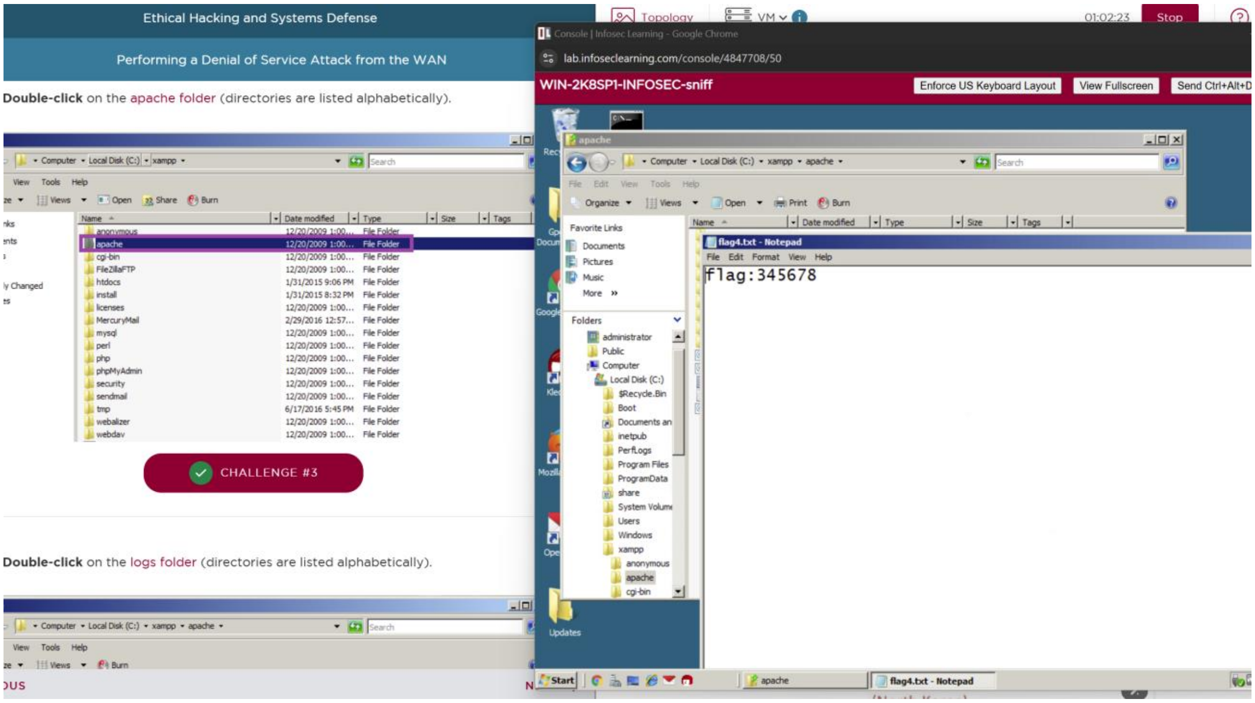
This picture shows the flag found in cat ip3.txt


Ethical Hacking and Systems Defense
Performing a Denial of Service Attack from the WAN
CHALLENGE #2
Topology
VM
00:41:57
Stop
lab.infoseclearning.com/console/4845479/50
WIN-2K8SP1-INFOSEC-sniff
Enforce US Keyboard Layout
View Fullscreen
Send Ctrl+Alt+Delete
xampp
Computer ▸ Local Disk (C:) ▸ xampp ▸
flag3.txt - Notepad
Start

on the apache folder (directories are listed alphabetically).

```
flag:774556
```

This screenshot shows the flag found in xampp folder

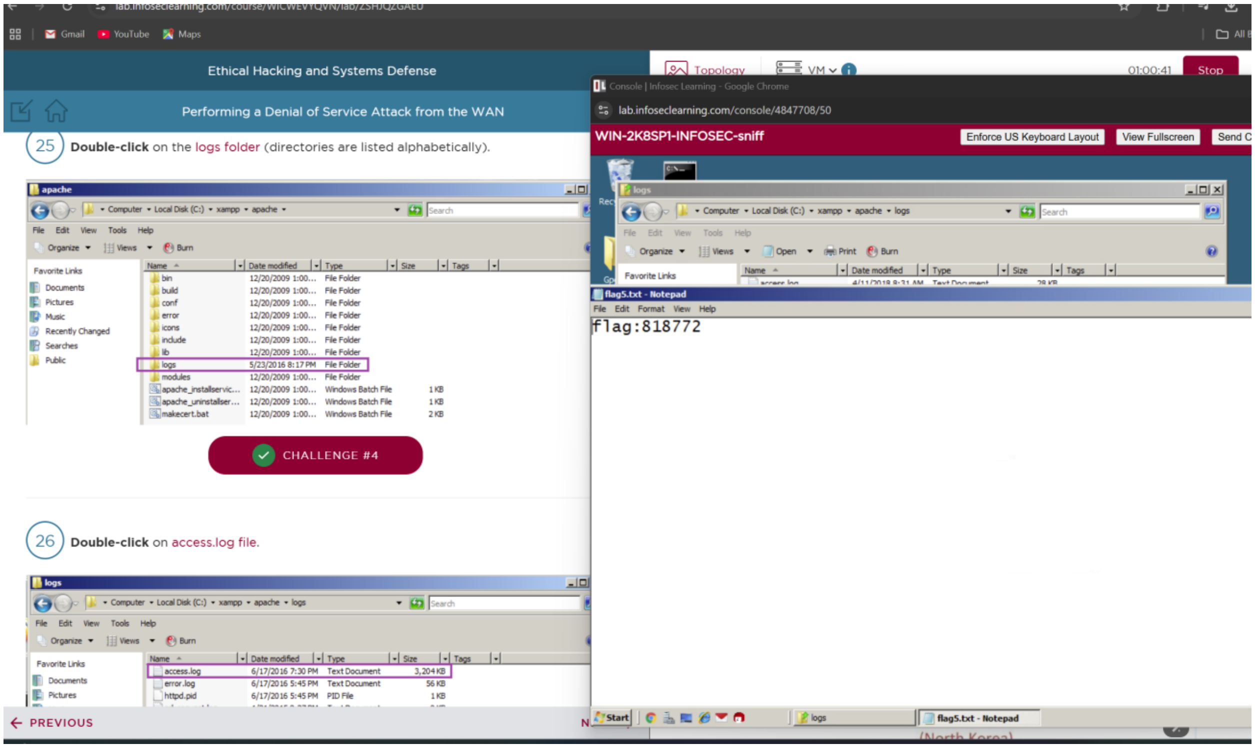

Ethical Hacking and Systems Defense
Performing a Denial of Service Attack from the WAN
CHALLENGE #3
Topology
VM
01:02:23
Stop
lab.infoseclearning.com/console/4847708/50
WIN-2K8SP1-INFOSEC-sniff
Enforce US Keyboard Layout
View Fullscreen
Send Ctrl+Alt+D
apache
Computer ▸ Local Disk (C:) ▸ xampp ▸ apache ▸
flag4.txt - Notepad
Start

Double-click on the apache folder (directories are listed alphabetically).

Double-click on the logs folder (directories are listed alphabetically).

```
flag:345678
```

This screenshot shows the flag found in apache folder



Ethical Hacking and Systems Defense

Performing a Denial of Service Attack from the WAN

25 **Double-click** on the logs folder (directories are listed alphabetically).

apache — Computer ▸ Local Disk (C:) ▸ xampp ▸ apache

Name	Date modified	Type	Size
bin	12/20/2009 1:00...	File Folder	
build	12/20/2009 1:00...	File Folder	
conf	12/20/2009 1:00...	File Folder	
error	12/20/2009 1:00...	File Folder	
icons	12/20/2009 1:00...	File Folder	
include	12/20/2009 1:00...	File Folder	
lib	12/20/2009 1:00...	File Folder	
logs	5/23/2016 8:17 PM	File Folder	
modules	12/20/2009 1:00...	File Folder	
apache_installservic...	12/20/2009 1:00...	Windows Batch File	1 KB
apache_uninstallser...	12/20/2009 1:00...	Windows Batch File	1 KB
makecert.bat	12/20/2009 1:00...	Windows Batch File	2 KB

CHALLENGE #4

26 **Double-click** on access.log file.

logs — Computer ▸ Local Disk (C:) ▸ xampp ▸ apache ▸ logs

Name	Date modified	Type	Size
access.log	6/17/2016 7:30 PM	Text Document	3,204 KB
error.log	6/17/2016 5:45 PM	Text Document	56 KB
httpd.pid	6/17/2016 5:45 PM	PID File	1 KB

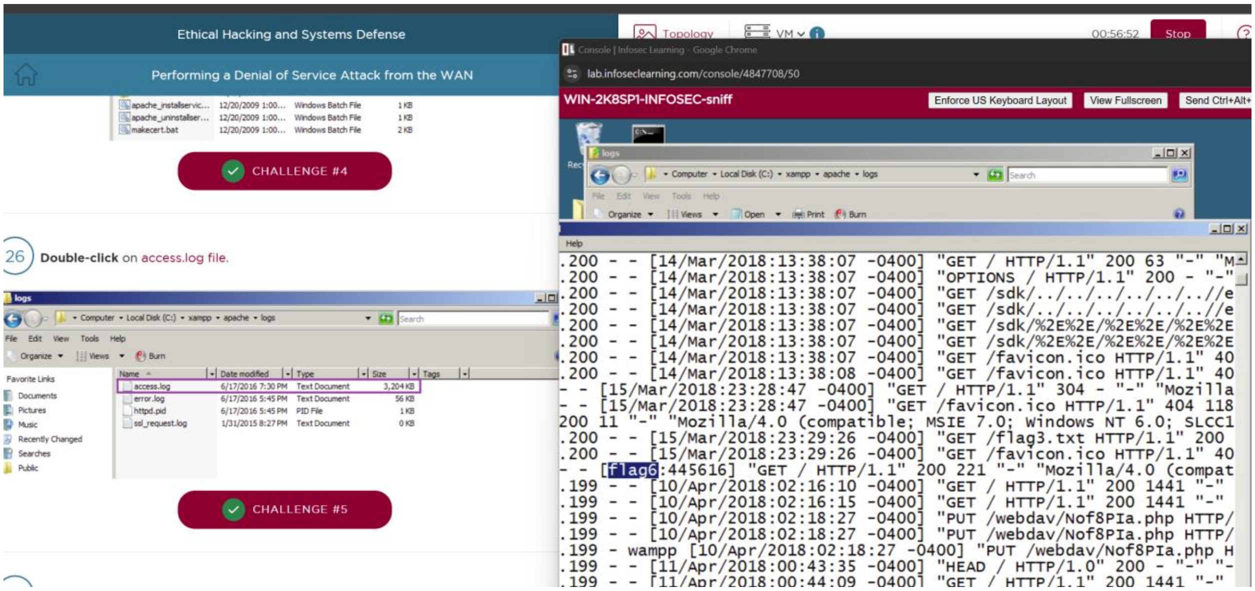
WIN-2K8SP1-INFOSEC-sniff

flag5.txt - Notepad

```
flag:818772
```

lab.infoseclearning.com/course/WICWEVYQVN/lab/ZSHJQZGAEU
Gmail
YouTube
Maps
Topology
VM
01:00:41
Stop
Console | Infosec Learning - Google Chrome
lab.infoseclearning.com/console/4847708/50
Enforce US Keyboard Layout
View Fullscreen
PREVIOUS
Start

This screenshot shows the flag found in logs folder



Ethical Hacking and Systems Defense

Performing a Denial of Service Attack from the WAN

CHALLENGE #4

26 **Double-click** on access.log file.

logs — Computer ▸ Local Disk (C:) ▸ xampp ▸ apache ▸ logs

Name	Date modified	Type	Size
access.log	6/17/2016 7:30 PM	Text Document	3,204 KB
error.log	6/17/2016 5:45 PM	Text Document	56 KB
httpd.pid	6/17/2016 5:45 PM	PID File	1 KB
ssl_request.log	1/31/2015 8:27 PM	Text Document	0 KB

CHALLENGE #5

WIN-2K8SP1-INFOSEC-sniff

```
.200 - - [14/Mar/2018:13:38:07 -0400] "GET / HTTP/1.1" 200 63 "-" "M
.200 - - [14/Mar/2018:13:38:07 -0400] "OPTIONS / HTTP/1.1" 200 - "-"
.200 - - [14/Mar/2018:13:38:07 -0400] "GET /sdk/../../../../../..//e
.200 - - [14/Mar/2018:13:38:07 -0400] "GET /sdk/../../../../../..//e
.200 - - [14/Mar/2018:13:38:07 -0400] "GET /sdk/%2E%2E/%2E%2E/%2E%2E
.200 - - [14/Mar/2018:13:38:07 -0400] "GET /sdk/%2E%2E/%2E%2E/%2E%2E
.200 - - [14/Mar/2018:13:38:07 -0400] "GET /favicon.ico HTTP/1.1" 40
.200 - - [14/Mar/2018:13:38:08 -0400] "GET /favicon.ico HTTP/1.1" 40
- - [15/Mar/2018:23:28:47 -0400] "GET / HTTP/1.1" 304 - "-" "Mozilla
- - [15/Mar/2018:23:28:47 -0400] "GET /favicon.ico HTTP/1.1" 404 118
200 11 "-" "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; SLCC1
.200 - - [15/Mar/2018:23:29:26 -0400] "GET /flag3.txt HTTP/1.1" 200
.200 - - [15/Mar/2018:23:29:26 -0400] "GET /favicon.ico HTTP/1.1" 40
- - [flag6:445616] "GET / HTTP/1.1" 200 221 "-" "Mozilla/4.0 (compat
.199 - - [10/Apr/2018:02:16:10 -0400] "GET / HTTP/1.1" 200 1441 "-"
.199 - - [10/Apr/2018:02:16:15 -0400] "GET / HTTP/1.1" 200 1441 "-"
.199 - - [10/Apr/2018:02:18:27 -0400] "PUT /webdav/Nof8PIa.php HTTP/
.199 - - [10/Apr/2018:02:18:27 -0400] "PUT /webdav/Nof8PIa.php HTTP/
.199 - wampp [10/Apr/2018:02:18:27 -0400] "PUT /webdav/Nof8PIa.php H
.199 - - [11/Apr/2018:00:43:35 -0400] "HEAD / HTTP/1.0" 200 - "-" "-
.199 - - [11/Apr/2018:00:44:09 -0400] "GET / HTTP/1.1" 200 1441 "-"
```

Topology
VM
00:56:52
Stop
Console | Infosec Learning - Google Chrome
lab.infoseclearning.com/console/4847708/50
Enforce US Keyboard Layout
View Fullscreen
Send Ctrl+Alt+
Help

This screenshot shows the flag found in access.log folder

Conclusion & Wrap-Up

Summary with:

Observations

During this lab, I observed how each type of denial-of-service (DoS) flood—TCP, UDP, and HTTP—generated significant traffic on the network, affecting the target's ability to respond to requests. The TCP and UDP floods quickly saturated the network bandwidth, while the HTTP flood created a massive spike in traffic that disrupted web services specifically. Packet analysis with tcpdump and capinfos clearly demonstrated the immense load these attacks placed on the network in a short period, highlighting the resource consumption associated with DoS attacks

Identified risks

The primary risk identified was the ease with which an attacker can overwhelm services and exhaust network resources, potentially leading to extended downtime and compromised availability of critical applications. The attacks also showed that, depending on the protocol, various aspects of the network stack are vulnerable. For instance, a TCP flood targets the connection-handling capacity, while HTTP floods exploit web service responsiveness, indicating that comprehensive defense strategies are needed across all layers**.**

Suggested recommendations

To mitigate these types of attacks, I would recommend implementing rate limiting, establishing firewall rules to detect and block excessive requests, and using Intrusion Detection and Prevention Systems (IDPS) to flag unusual traffic patterns. Implementing a Web Application Firewall (WAF) specifically for HTTP traffic could also reduce the impact of HTTP floods. Network segmentation and using redundant resources may also help maintain service availability during such attacks.

Your successes & failures

I successfully set up each type of flood attack, capturing and analyzing traffic with tcpdump and capinfos. However, I initially struggled with configuring the network interface for packet capture on the sniffer machine, which delayed the setup. Once resolved, the packet capture and flood attacks went smoothly, though managing and interpreting the high volume of data generated by the attacks required careful handling**.**

Challenges

One significant challenge was managing the immense amount of traffic generated during each attack, as the network became quickly congested. This led to some delays in packet capture and analysis, highlighting the potential for real-world attacks to slow down even monitoring systems. Additionally, interpreting the data to quantify the impact on specific services required detailed analysis, especially when discerning differences in impact between TCP, UDP, and HTTP floods.

- This table prioritize remediation steps based on the potential impact of each risk on network and service performance.

Risk	Description	Risk Priority	Remediation
Network Bandwidth Exhaustion	High volume of packets floods the network, consuming all available bandwidth and slowing down services.	High	Implement rate limiting and configure firewall rules to detect and block abnormal traffic patterns.
Service Downtime	Targeted services (TCP, UDP, HTTP) become unresponsive, affecting application availability.	High	Use Intrusion Detection and Prevention Systems (IDPS) to identify DoS patterns and dynamically adjust network traffic.
Connection Saturation (TCP Flood)	The TCP flood overloads the target's connection capacity, blocking legitimate connections.	Medium	Deploy SYN cookies to manage excessive TCP requests and configure the firewall to limit incoming TCP connections.
Application Layer Exhaustion (HTTP Flood)	Web services have become unresponsive due to the high HTTP traffic volume.	Medium	Utilize a Web Application Firewall (WAF) to manage excessive HTTP requests and restrict HTTP requests based on origin.
Resource Exhaustion on Monitoring Systems	High traffic load can overwhelm network monitoring tools, hindering traffic analysis.	Low	Implement resource scaling for monitoring systems and apply filters to capture relevant traffic without saturation.