



The University of Texas at San Antonio™

The Cyber Center for Security and Analytics



UNIVERSITY OF
SOUTH CAROLINA

ZEEK INTRUSION DETECTION SERIES

Lab 5: Generating, Capturing and Analyzing DoS and DDoS-centric Network Traffic

Document Version: **03-13-2020**



Award 1829698

“CyberTraining CIP: Cyberinfrastructure Expertise on High-throughput
Networks for Big Science Data Transfers”

Contents

Overview	4
Objective	4
Lab topology.....	4
Lab settings	4
Lab roadmap	5
1 Introduction to DoS and DDoS activity	5
1.1 DoS attack characteristics	5
1.2 DDoS attack characteristics.....	6
2 Generating real-time DoS traffic.....	6
2.1 Starting a new instance of Zeek	6
2.2 Launching Mininet.....	7
2.3 Setting up the zeek2 machine for live network capture.....	9
2.4 Launching LOIC.....	10
2.5 Using the zeek1 virtual machine to launch a TCP-based DoS attack.....	12
2.6 Using the zeek1 virtual machine to launch a UDP-based DoS attack.....	13
3 Analyzing collected network traffic	16
3.1 Analyzing TCP-based traffic.....	16
3.1.1 TCP Example Query 1.....	17
3.1.2 TCP Example Query 2.....	17
3.2 Analyzing UDP-based traffic.....	18
3.3 Closing the current instance of Zeek.....	19
References	20

Overview

This lab covers Denial of Service (DoS)-based network traffic. The lab introduces the generation of DoS-based traffic for testing purposes and uses Zeek to process the collected traffic.

Objective

By the end of this lab, students should be able to:

1. Generate real-time DoS and DDoS traffic.
2. Experiment with the Low Orbit Ion Canon (LOIC) software.
3. Analyze collected DDoS traffic.

Lab topology

Figure 1 shows the lab workspace topology. This lab primarily uses the *zeek1* virtual machine to generate DoS-based traffic, and the *zeek2* virtual machine to perform live network capture.

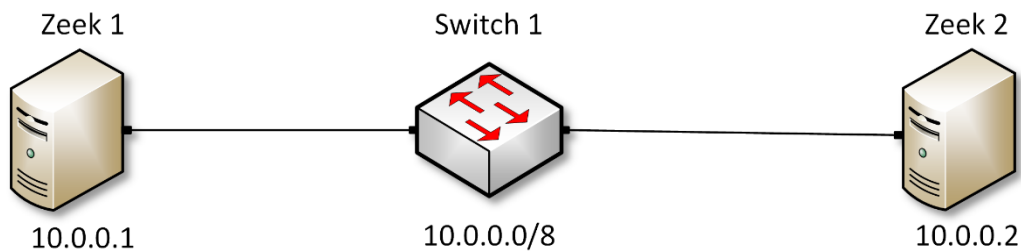


Figure 1. Lab topology.

Lab settings

The information (case-sensitive) in the table below provides the credentials necessary to access the machines used in this lab.

Table 1. Credentials to access the Client machine

Device	Account	Password
Client	admin	password

Table 2. Shell variables and their corresponding absolute paths.

Variable Name	Absolute Path
\$ZEEK_INSTALL	/usr/local/zeek
\$ZEEK_TESTING_TRACES	/home/zeek/zeek/testing/btest/Traces
\$ZEEK_PROTOCOLS_SCRIPT	/home/zeek/zeek/scripts/policy/protocols

Lab roadmap

This lab is organized as follows:

1. Section 1: Introduction to DoS and DDoS activity.
2. Section 2: Generating real-time DoS traffic.
3. Section 3: Analyzing collected network traffic.

1 Introduction to DoS and DDoS activity

Denial-of-Service (DoS) is an attack launched by a malicious user to render a target machine or network resource unavailable to its intended users. Distributed Denial-of-Service (DDoS) is an attack originated from different sources to flood the victim's resources. A DDoS attack is more effective than a normal DoS and is harder to mitigate since unlike DoS, it is impossible to stop the attack simply by blocking a single source.

The different types of DoS attacks can be grouped by the traffic they generate, the bandwidth they consume, the services they disrupt, etc. Traffic-based DoS attacks aim at flooding the target with a large volume unsolicited traffic. Bandwidth-based DoS attacks involve transmitting a massive amount of junk data to overload the victim and render its network equipment congested.

1.1 DoS attack characteristics

DoS attacks generally involve flooding a targeted victim with network traffic to cause a crash and make it unavailable to benign users. In this lab we explore two common DoS attacks:

- **SYN flood**: an attacker attempts to overwhelm the server machine by sending a constant stream of TCP connection requests, forcing the server to allocate resources for each new connection until all resources are exhausted¹.
- **ICMP flood**: the attacker abuses ICMP **ping** and floods the victim computer with Echo Request messages. When a computer receives an ICMP Echo Request message it responds with an ICMP Echo Reply message².

1.2 DDoS attack characteristics

DDoS attacks involve using a large number of devices to flood a victim. With an increased number of exploited machines, the amount of resources available to the attacker is far higher. Some relevant DDoS attacks are:

- **HTTP flood**: simple attack but requires a large number of resources. An attacker who controls several devices (botnet) can continually flood a server with HTTP requests until the server becomes unavailable and unable to respond to additional incoming requests.
- **SYN flood**: similar to the DoS SYN flood, a botnet initiates several sessions without completing a TCP handshake, causing the victim to consume its available resources.
- **Amplification attack**: attackers abuse UDP-based network protocols to launch DDoS attacks that exceed hundreds of Gbps in traffic volume. This is achieved via reflective DDoS attacks where an attacker does not directly send traffic to the victim but sends spoofed network packets to a large number of systems that reflect the traffic to the victim³. Domain Name System (DNS) and Network Time Protocol (NTP) are examples of application-layer protocols that act as potential amplification attack vectors.

DoS and DDoS attacks can cause catastrophic fallout and monetary losses to a victim.

2 Generating real-time DoS traffic

This lab uses the Low Orbit Ion Canon (LOIC), open-source network stress testing and DoS attack generator. LOIC can be found in the following Github repository. To access the following link, users must have access to an external computer connected to the Internet, because the Zeek Lab topology does not have an active Internet connection.

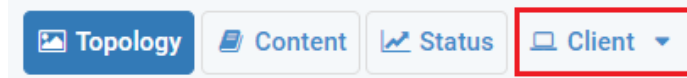
```
https://github.com/NewEraCracker/LOIC
```

Similar to the **nmap** utility, **LOIC** can be used to replicate DoS or DDoS activity for testing purposes. **LOIC** has a Graphical User Interface (GUI), which facilitates the attack's customization.

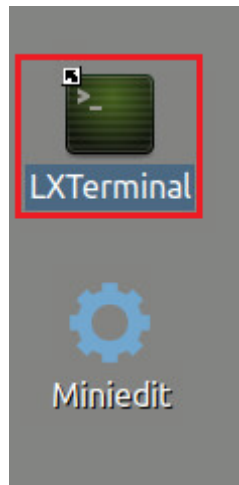
In this lab, Zeek's default packet capture processing will generate log files containing organized network traffic statistics. In this section, *zeek2* virtual machine is used for live capture and *zeek1* virtual machine is used to generate DoS-related traffic.

2.1 Starting a new instance of Zeek

Step 1. From the top of the screen, click on the *Client* button as shown below to enter the *Client* machine.



Step 2. The *Client* machine will now open, and the desktop will be displayed. On the left side of the screen, click on the LXTerminal icon as shown below.



Step 3. Start Zeek by entering the following command on the terminal. This command enters Zeek's default installation directory and invokes `zeekctl` tool to start a new instance. To type capital letters, it is recommended to hold the `Shift` key while typing rather than using the `Caps` key. When prompted for a password, type `password` and hit `Enter`.

```
cd $ZEEK_INSTALL/bin && sudo ./zeekctl start
```

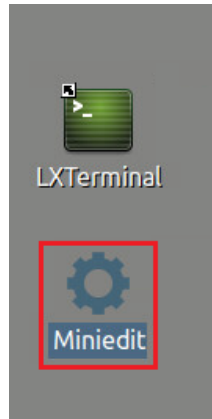
```
zeek@admin: /usr/local/zeek/bin
File Edit Tabs Help
zeek@admin:~$ cd $ZEEK_INSTALL/bin && sudo ./zeekctl start
[sudo] password for zeek:
starting zeek ...
zeek@admin: /usr/local/zeek/bin$
```

A new instance of Zeek is now active, and we are ready to proceed to the next section of the lab.

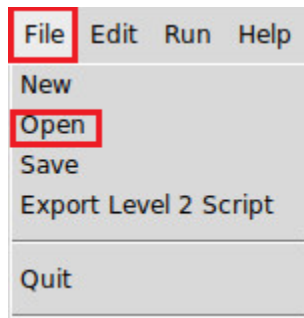
If you see error messages during the new Zeek instance initializing process, please ignore it.

2.2 Launching Mininet

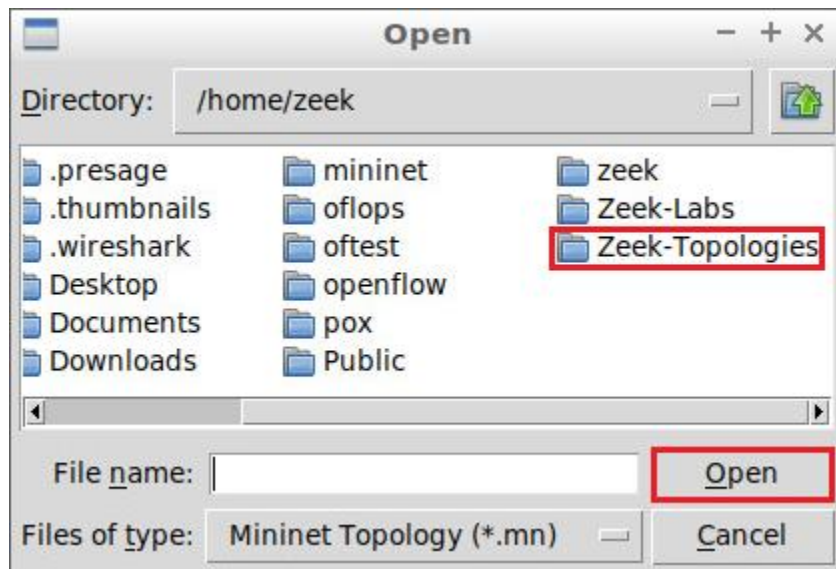
Step 1. From the *Client* machine's desktop, on the left side of the screen, click on the MiniEdit icon as shown below. When prompted for a password, type `password` and hit `Enter`. The MiniEdit editor will now launch.



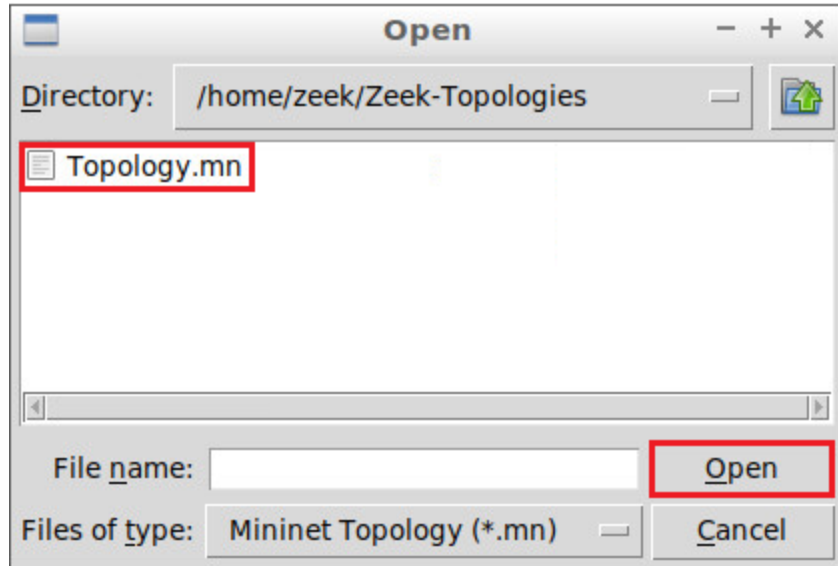
Step 2. The MiniEdit editor will now launch and allow for the creation of new, virtualized lab topologies. Load the correct topology by clicking the `Open` button within the `File` tab on the top left of the MiniEdit editor.



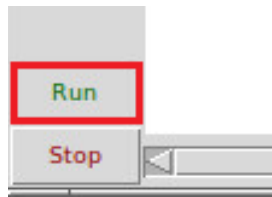
Step 3. Navigate to the Zeek-Topologies directory by scrolling to the right of the active directories and double clicking the Zeek-Topologies icon, or by clicking the `Open` button.



Step 4. Select the `Topology.mn` file by double clicking the `Topologies.mn` icon, or by clicking the `Open` button.

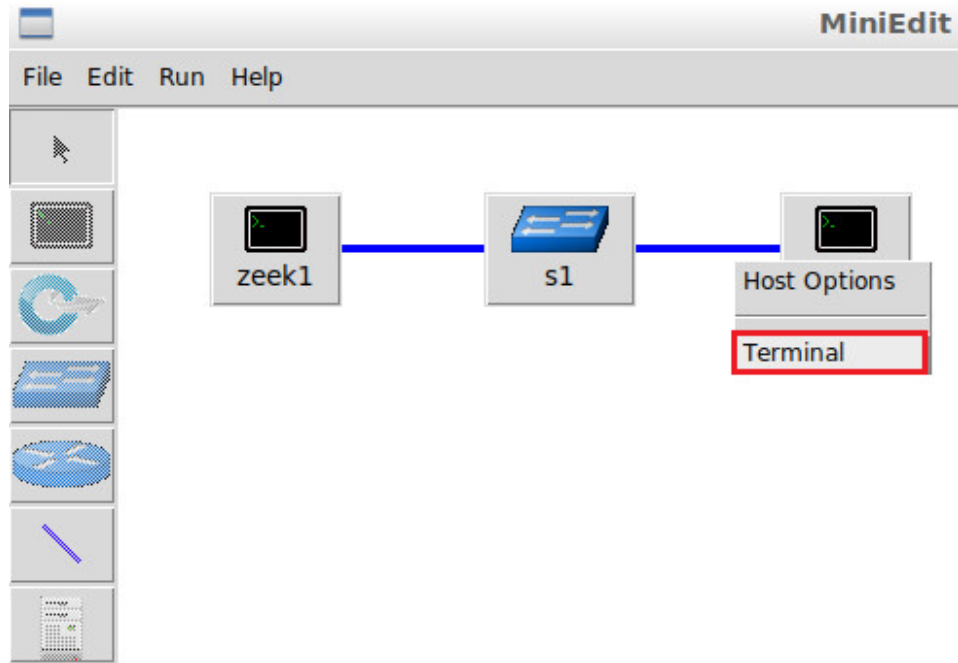


Step 5. To begin running the virtual machines, navigate to the **Run** button, found on the bottom left of the Miniedit editor, and select the **Run** button, as seen in the image below.



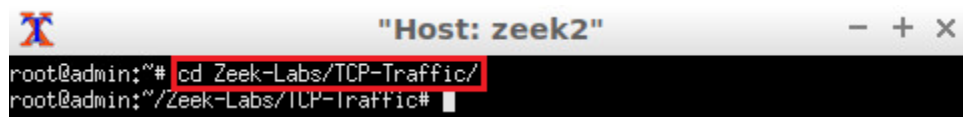
2.3 Setting up the zeek2 machine for live network capture

Step 1. Launch the *zeek2* terminal by holding the right mouse button on the desired machine and clicking the *Terminal* button.



Step 2. From the *zeek2* terminal, navigate to the TCP-Traffic directory.

```
cd Zeek-Labs/TCP-Traffic/
```



Step 3. Start live packet capture on interface *zeek2-eth0* and save the output to a file named *tcptraffic.pcap*.

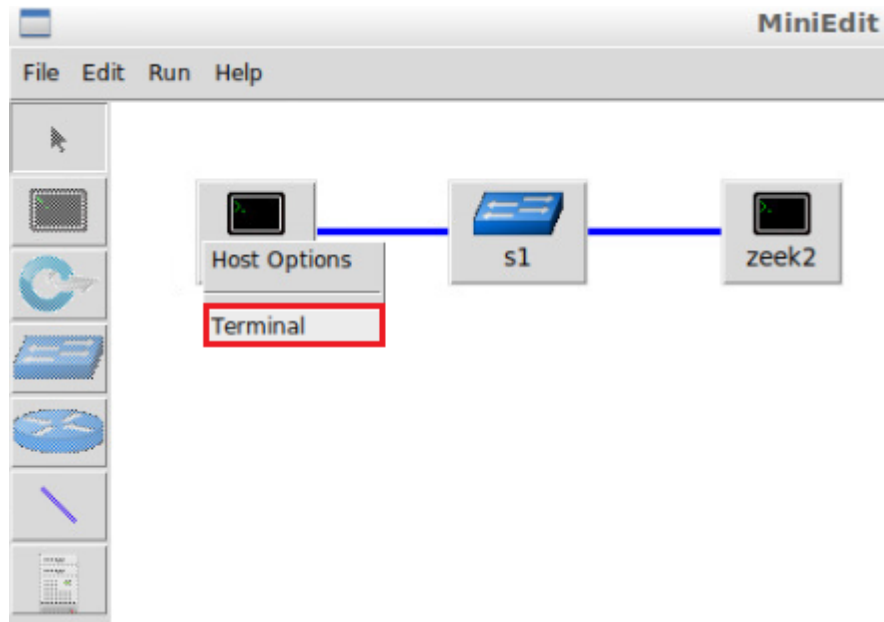
```
tcpdump -i zeek2-eth0 -s 0 -w tcptraffic.pcap
```



The *zeek2* virtual machine is now ready to begin collecting live network traffic. Next, we will use the *zeek1* machine to generate scan-based network traffic.

2.4 Launching LOIC

Step 1. Minimize the *zeek2* Terminal and open the *zeek1* Terminal by following the previous steps. If necessary, right click within the Miniedit editor to activate your cursor.



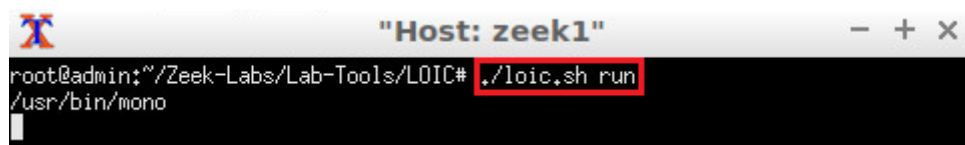
Step 2. Navigate to the *Zeek-Labs/Lab-Tools/LOIC* directory.

```
cd Zeek-Labs/Lab-Tools/LOIC
```

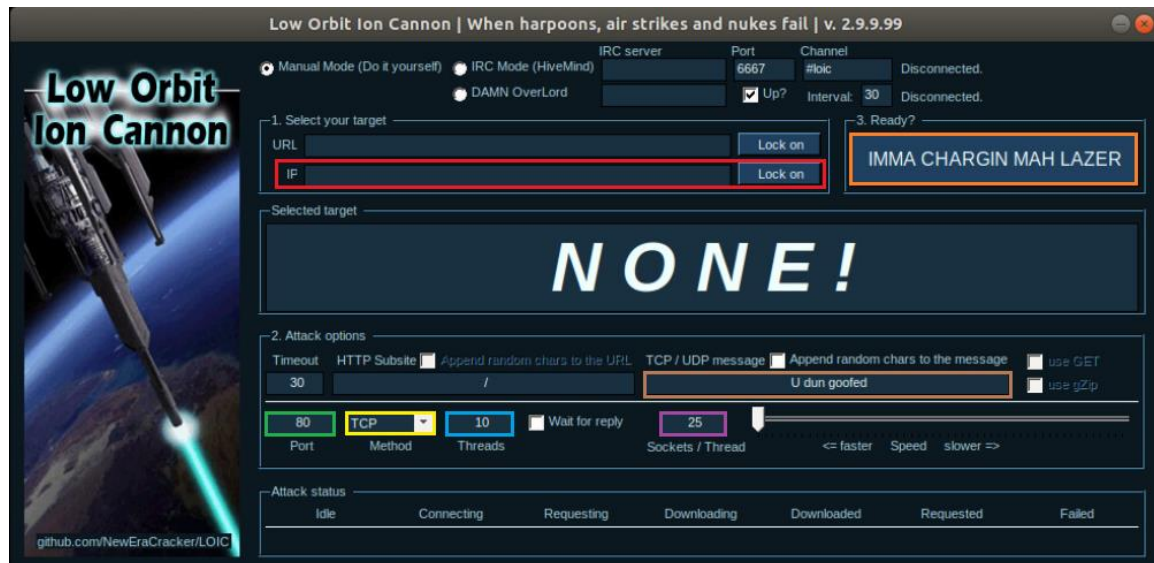


Step 3. Execute the *loic.sh* shell script by entering the following command in the terminal.

```
./loic.sh run
```



Step 4. View the LOIC GUI. If necessary, scale the GUI to a smaller size to fit on the *zeek1* virtual machine's display.



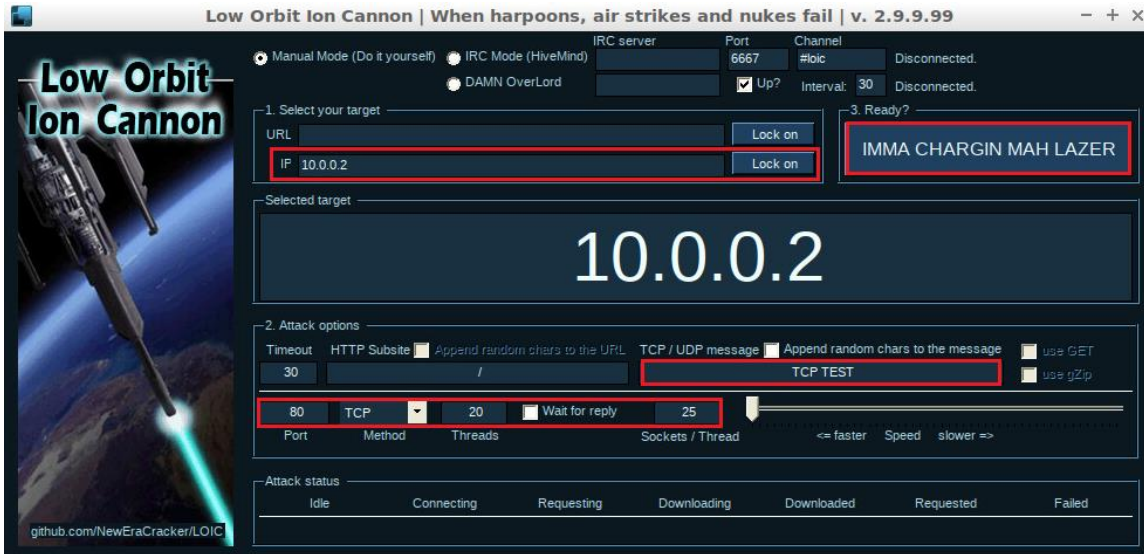
The figure above shows the LOIC interface. Important features highlighted with colored boxes are explained as follows:

1. **Red Box**: target IP address. After entering an IP address, clicking the *Lock on* button will select the IP as the target destination address.
2. **Green Box**: target port. Can be changed depending on which method is used to launch the DoS attack.
3. **Yellow Box**: target method. Can be changed to define which protocol is used to launch the DoS attack.
4. **Blue Box**: number of threads. Indicates the amount of resources *LOIC* will allocate on the host machine.
5. **Purple Box**: number of sockets per thread. Increasing the number of sockets per thread will exponentially increase the speed of the DoS attack; however, it also requires more resources on the host machine.
6. **Brown Box**: packet payload. Used to define what each packet will contain as payload.
7. **Orange Box**: start button. After customizing a desired attack, this button is used to launch the attack.

2.5 Using the zeek1 virtual machine to launch a TCP-based DoS attack

Step 1: Customize the DoS attack by entering the following values in their respective input boxes.

IP:	10.0.0.2
Port:	80
Method:	TCP
Threads:	20
Sockets:	25
Payload:	TCP TEST

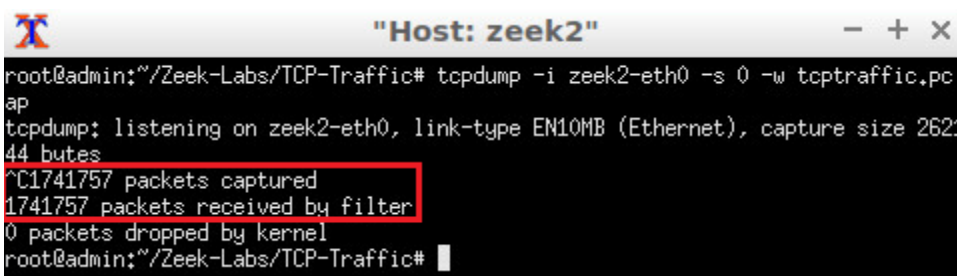


Step 2. Click the *Lock on* button to save the current configurations. Click the **Start** (*IMMA CHARGIN MAH LAZER*) button to begin the DoS attack. Wait roughly 10 seconds and click the **Stop** (*Stop flooding*) button to stop the DoS attack.

Step 3. Minimize the *zeek1 Terminal* and open the *zeek2 Terminal* using the navigation bar at the bottom of the screen. If necessary, right click within the Miniedit editor to activate your cursor.

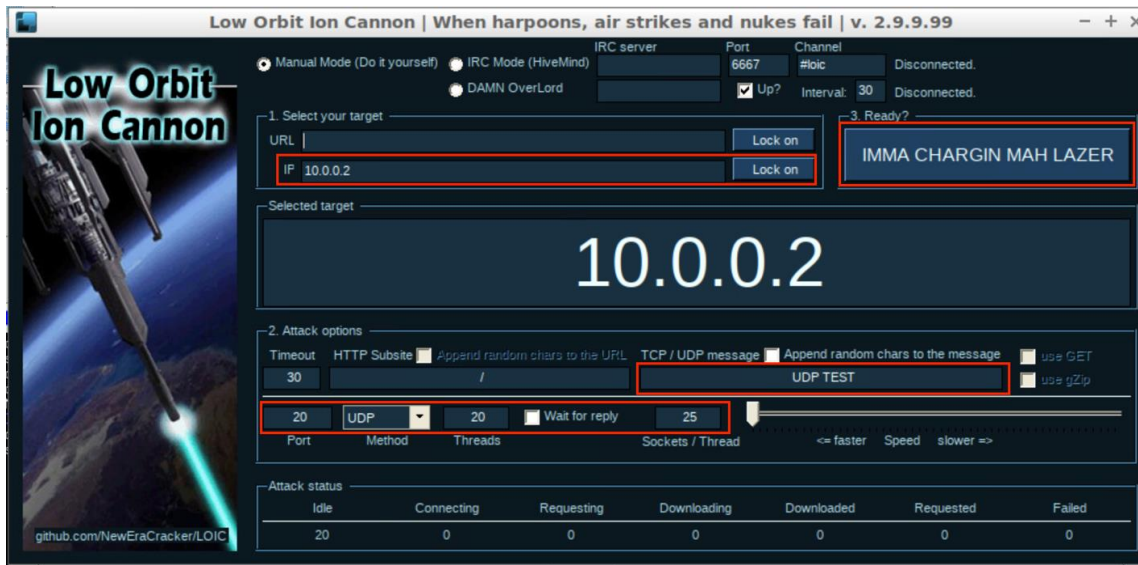


Step 4. Use the **Ctrl+c** key combination to stop live traffic capture. Statistics of the capture session will be displayed with network packets being stored in the new *tcptraffic.pcap* file.



Within the 10 seconds timeframe, 1,741,757 packets were generated and collected. This number of packets verifies that DoS attacks generate an immense amount of network traffic and can be compared against the much smaller number of packets generated during the previous scan events.

2.6 Using the *zeek1* virtual machine to launch a UDP-based DoS attack

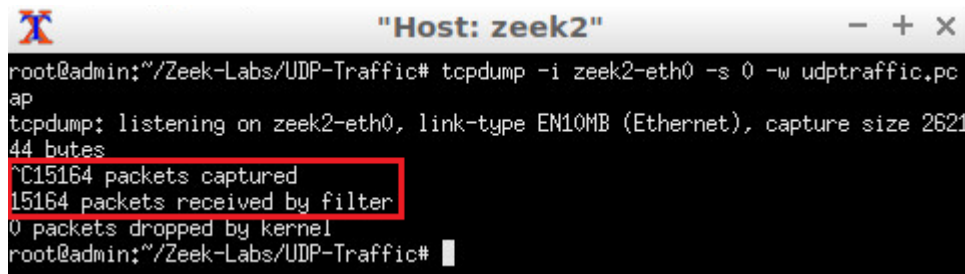


Step 5. Click the *Lock on* button to save the current configurations. Click the *Start (IMMA CHARGIN MAH LAZER)* button to begin the DoS attack. Wait for 10 seconds and click the *Stop (Stop flooding)* button to stop the DoS attack.

Step 6. Minimize the *zeek1 Terminal* and open the *zeek2 Terminal* using the navigation bar at the bottom of the screen. If necessary, right click within the Miniedit editor to activate your cursor.

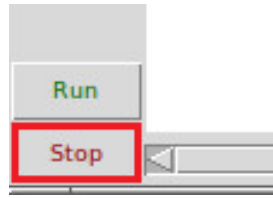


Step 7. Use the `Ctrl+c` key combination to stop live traffic capture. Statistics of the capture session will be displayed. 15,164 packets were recorded by the interface, which were then captured and stored in the new *tcptraffic.pcap* file.



While the UDP-based DoS attack did not generate as much network traffic as the TCP-based DoS attack, heavy amounts of traffic were generated by a single machine. Scaled to a large-scale attack, DoS attacks are extremely debilitating.

Step 8. Stop the current Mininet session by clicking the *Stop* button on the bottom left of the MiniEdit editor, and close the MiniEdit editor by clicking the `⌘` on the top right of the editor.



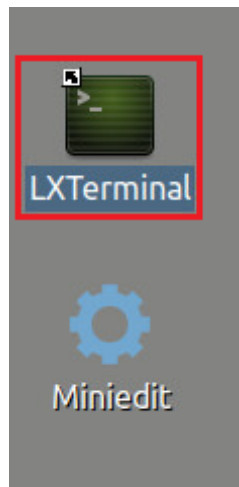
We will now return to the *Client* machine to process and analyze the newly generated network traffic.

3 Analyzing collected network traffic

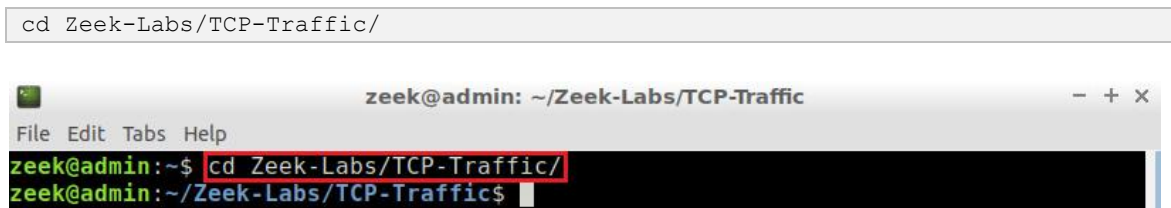
After successfully conducting both a TCP-based and UDP-based DoS attack, we can begin to analyze the collected network traffic using Zeek and the `zeek-cut` utility commands to display the capture traffic.

3.1 Analyzing TCP-based traffic

Step 1. On the left side of the *Client* desktop, click on the LXTerminal icon as shown below.



Step 2. Navigate to the *TCP-Traffic* directory to find the *tcptraffic.pcap* file.



Step 3. View the file contents of the *TCP-Traffic* directory to ensure that the *tcptraffic.pcap* file was successfully saved.




```

zeek@admin: ~/Zeek-Labs/TCP-Traffic
File Edit Tabs Help
zeek@admin:~/Zeek-Labs/TCP-Traffic$ ls
tcptraffic.pcap
zeek@admin:~/Zeek-Labs/TCP-Traffic$

```

Step 4. Use the following Zeek command to process the packet capture file.

```
zeek -C -r tcptraffic.pcap
```

```

zeek@admin: ~/Zeek-Labs/TCP-Traffic
File Edit Tabs Help
zeek@admin:~/Zeek-Labs/TCP-Traffic$ zeek -C -r tcptraffic.pcap
zeek@admin:~/Zeek-Labs/TCP-Traffic$

```

Step 5. List the generated Zeek log files.

```
ls
```

```

zeek@admin: ~/Zeek-Labs/TCP-Traffic
File Edit Tabs Help
zeek@admin:~/Zeek-Labs/TCP-Traffic$ ls
conn.log packet_filter.log tcptraffic.pcap
zeek@admin:~/Zeek-Labs/TCP-Traffic$

```

3.1.1 TCP Example Query 1

Example 1: Show the source IP addresses that generated the most network traffic, organized in descending order.

```
zeek-cut id.resp_h < conn.log | sort | uniq -c | sort -rn | head -n 10
```

```

zeek@admin: ~/Zeek-Labs/TCP-Traffic
File Edit Tabs Help
zeek@admin:~/Zeek-Labs/TCP-Traffic$ zeek-cut id.resp_h < conn.log | sort | uniq -c | sort -rn | head -n 10
870871 10.0.0.2
8 ff02::2
zeek@admin:~/Zeek-Labs/TCP-Traffic$

```

The *zeek2* virtual machine received 870,871 TCP packets. This command, or a similar one, can be useful in real-world environments to detect vulnerable hosts within a network – allowing for the process of securing and mitigating possible threats.

3.1.2 TCP Example Query 2

Example 1: Show the destination ports that received the most traffic, organized in descending order.

```
zeek-cut id.resp_p < conn.log | sort | uniq -c | sort -rn | head -n 10
```

```
zeek@admin: ~/Zeek-Labs/TCP-Traffic
File Edit Tabs Help
zeek@admin:~/Zeek-Labs/TCP-Traffic$ zeek-cut id.resp_p < conn.log | sort | uniq -c | sort -rn | head -n 10
870871 80
8 134
zeek@admin:~/Zeek-Labs/TCP-Traffic$
```

We can see that 870,871 packets were received by the *zeek2* virtual machine on port 80, which is the port we specified for the *zeek1* virtual machine to target. Additional ports may be discovered during processing, slightly variable due to LOIC attempting to establish connections; however, it is clear the most targeted port is the one we specified in the DoS attack.

3.2 Analyzing UDP-based traffic

Step 1. Navigate to the *UDP-Traffic* directory to find the *udptraffic.pcap* file.

```
cd ~/Zeek-Labs/UDP-Traffic/
```

```
zeek@admin: ~/Zeek-Labs/UDP-Traffic
File Edit Tabs Help
zeek@admin:~/Zeek-Labs/UDP-Traffic$ cd ~/Zeek-Labs/UDP-Traffic/
zeek@admin:~/Zeek-Labs/UDP-Traffic$
```

Step 2. View the file contents of the *TCP-Traffic* directory to ensure that the *udptraffic.pcap* file was successfully saved.

```
ls
```

```
zeek@admin: ~/Zeek-Labs/UDP-Traffic
File Edit Tabs Help
zeek@admin:~/Zeek-Labs/UDP-Traffic$ ls
udptraffic.pcap
zeek@admin:~/Zeek-Labs/UDP-Traffic$
```

Step 3. Use the following Zeek command to process the packet capture file.

```
zeek -C -r udptraffic.pcap
```

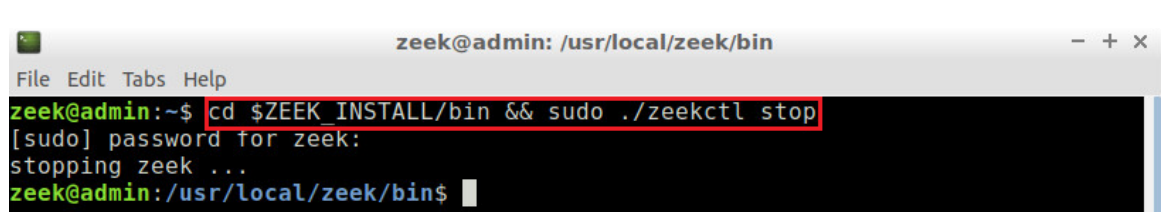
```
zeek@admin: ~/Zeek-Labs/UDP-Traffic
File Edit Tabs Help
zeek@admin:~/Zeek-Labs/UDP-Traffic$ zeek -C -r udptraffic.pcap
zeek@admin:~/Zeek-Labs/UDP-Traffic$
```

Step 4. List the generated Zeek log files.

After you have finished the lab, it is necessary to terminate the currently active instance of Zeek. Shutting down a computer while an active instance persists will cause Zeek to shut down improperly and may cause errors in future instances.

Step 1. Stop Zeek by entering the following command on the terminal. If required, type `password` as the password. If the Terminal session has not been terminated or closed, you may not be prompted to enter a password. To type capital letters, it is recommended to hold the `Shift` key while typing rather than using the `Caps` key.

```
cd $ZEEK_INSTALL/bin && sudo ./zeekctl stop
```



```
zeek@admin:~/usr/local/zeek/bin$ cd $ZEEK_INSTALL/bin && sudo ./zeekctl stop
[sudo] password for zeek:
stopping zeek ...
zeek@admin:~/usr/local/zeek/bin$
```

Concluding this lab, we have introduced DoS and DDoS events, as well as generated and captured DoS traffic in the lab workspace environment. Networks require some form of denial-of-service mitigation or prevention tools since attacks can devastate unsecured networks.

References

1. Lemon, Jonathan, "Resisting SYN flood DoS attacks with a SYN cache," In BSDCon, vol. 2002, pp. 89-97. 2002.
2. Junior, R. B., & Kumar, S. (2014), "Apple's lion vs microsoft's windows 7: comparing built-in protection against ICMP flood attacks," Journal of information security, 5(03), 123.
3. Kührer, M., Hupperich, T., Rossow, C., & Holz, T. (2014). "Exit from hell? reducing the impact of amplification DDoS attacks,". In 23rd {USENIX} Security symposium ({USENIX} Security 14) (pp. 111-125).
4. Fachkha, Claude, Elias Bou-Harb, and Mourad Debbabi. "Fingerprinting internet DNS amplification DDoS activities." *2014 6th International Conference on New Technologies, Mobility and Security (NTMS)*. IEEE, 2014.
5. Fachkha, Claude, Elias Bou-Harb, and Mourad Debbabi. "Towards a forecasting model for distributed denial of service activities." *2013 IEEE 12th International Symposium on Network Computing and Applications*. IEEE, 2013.