UTSA®

The University of Texas at San Antonio™

The Cyber Center for Security and Analytics

UNIVERSITY OF SOUTH CAROLINA

ZEEK INSTRUSION DETECTION SERIES

Lab 4: Generating, Capturing and Analyzing Network Scanner Traffic

Document Version: 03-13-2020

Award 1829698
“CyberTraining CIP: Cyberinfrastructure Expertise on High-throughput
Networks for Big Science Data Transfers”

Contents

Overview

This lab is designed to provide an in-depth guide to scanning and probing network traffic. The lab demonstrates the generation of scan-based traffic and uses Zeek to process the collected traffic.

Objective

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

1. Perform Internet scanning and probing events.
2. Utilize the Nmap software.
3. Generate and collect scan traffic.

Lab topology

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

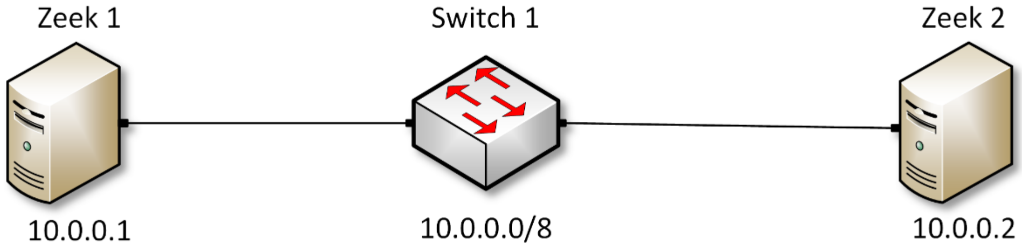


Figure 1. Lab topology.

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 Internet scanning and probing.
2. Section 2: Generating real time network scans.
3. Section 3: Analyzing collected network traffic.
4. Section 4: Detailing the importance of the Zeek interface topology.

1 Introduction to Internet scanning and probing

Internet scanning is the process of generating crafted traffic used to identify active devices on a network. A variety of software utilities and tools are used to replicate scan-related traffic for testing purposes. These crafted packets can be both stealthy and versatile. It is hard to determine scan-like activities when scanning traffic follows protocols' standards and specifications.

Malicious scanning is a reconnaissance technique used to collect information about a target's machine or network to facilitate an attack against it. Scanning is used by attackers to discover what ports are open, what services are running and identify system software, all to enable an attacker to more easily detect and exploit known vulnerabilities within a target machine[1].

This lab uses `nmap`, and its documentation can be found on the `nmap` website. 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://www.nmap.org/
```

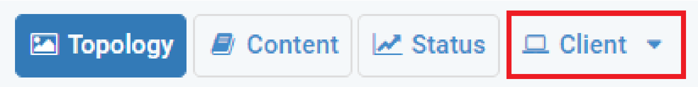
`nmap` has a wide array of scan-related functionalities such as the customization of a scan's transport protocol, ports, IP ranges, etc.

2 Generating real time network scans

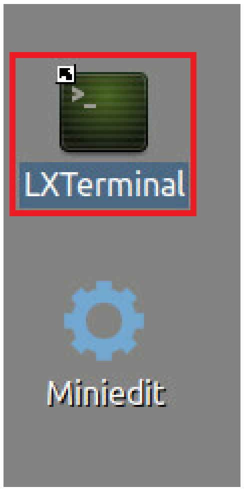
Zeek's default packet capture processing generates log files containing organized network traffic statistics. By leveraging the *zeek1* virtual machine to scan the *zeek2* virtual machine, we can better define and understand the steps it takes to both generate and capture scan traffic.

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.

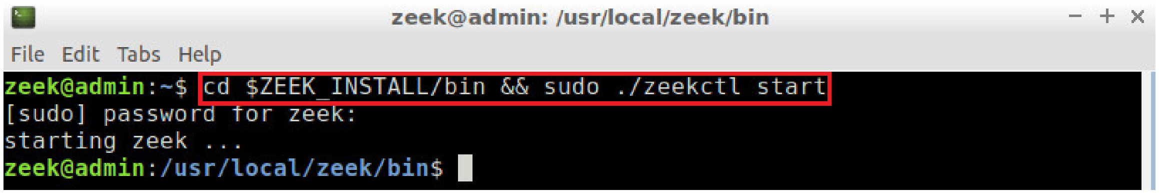


Step 2. The *Client* machine will now open, and the desktop will be displayed. On the left side of the screen, click on the LXTerminal icon as shown below.



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
```

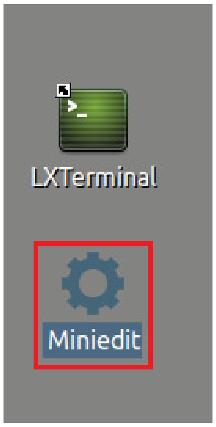


```
zeek@admin:~$ cd $ZEEK_INSTALL/bin && sudo ./zeekctl start
[sudo] password for zeek:
starting zeek ...
zeek@admin:/usr/local/zeek/bin$
```

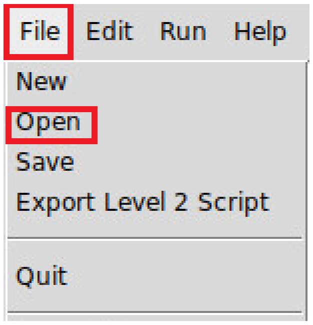
A new instance of Zeek is now active, and we are ready to proceed to the next section of the lab.

2.2 Launching Mininet

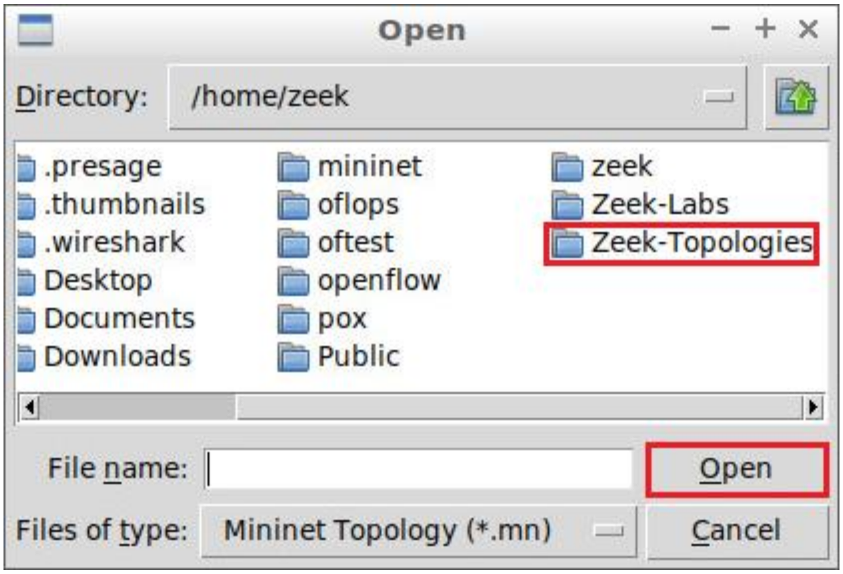
Step 1. From the *Client* machine's desktop, on the left side of the screen, click on the MiniEdit icon as shown below. When prompted for a password, type `password` and hit `Enter`. The MiniEdit editor will now launch.



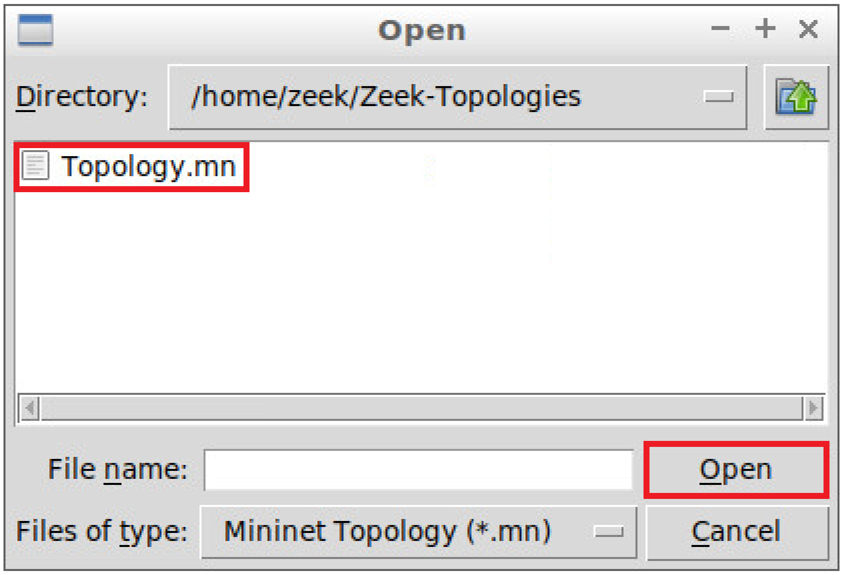
Step 2. The MiniEdit editor will now launch and allow for the creation of new, virtualized lab topologies. Load the correct topology by clicking the `Open` button within the `File` tab on the top left of the MiniEdit editor.



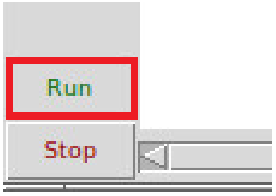
Step 3. Navigate to the Zeek-Topologies directory by scrolling to the right of the active directories and double clicking the Zeek-Topolgies icon, or by clicking the `Open` button.



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

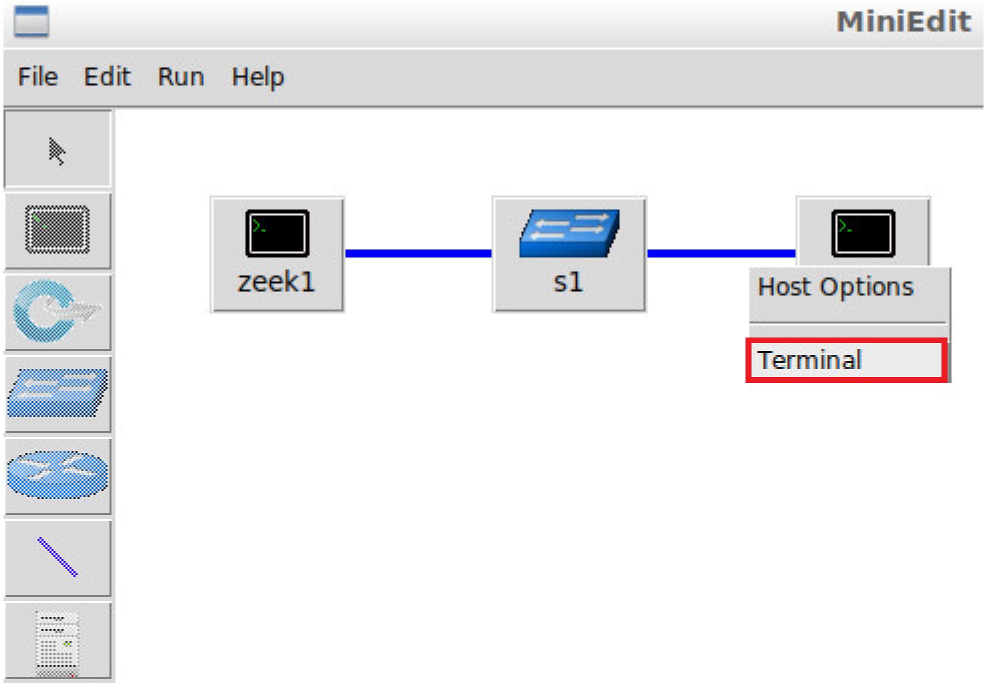

Open
Directory: /home/zeek/Zeek-Topologies
Topology.mn
File name:
Open
Files of type: Mininet Topology (*.mn)
Cancel

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.



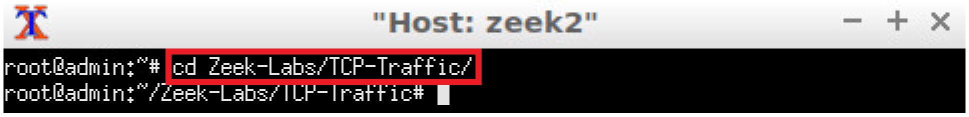
2.3 Setting up the zeek2 virtual machine for live network capture

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.

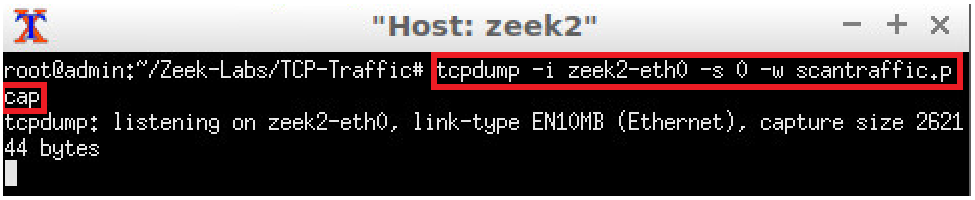
```
cd Zeek-Labs/TCP-Traffic/
```



```
root@admin:~# cd Zeek-Labs/TCP-Traffic/
root@admin:~/Zeek-Labs/TCP-Traffic# 
```

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

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

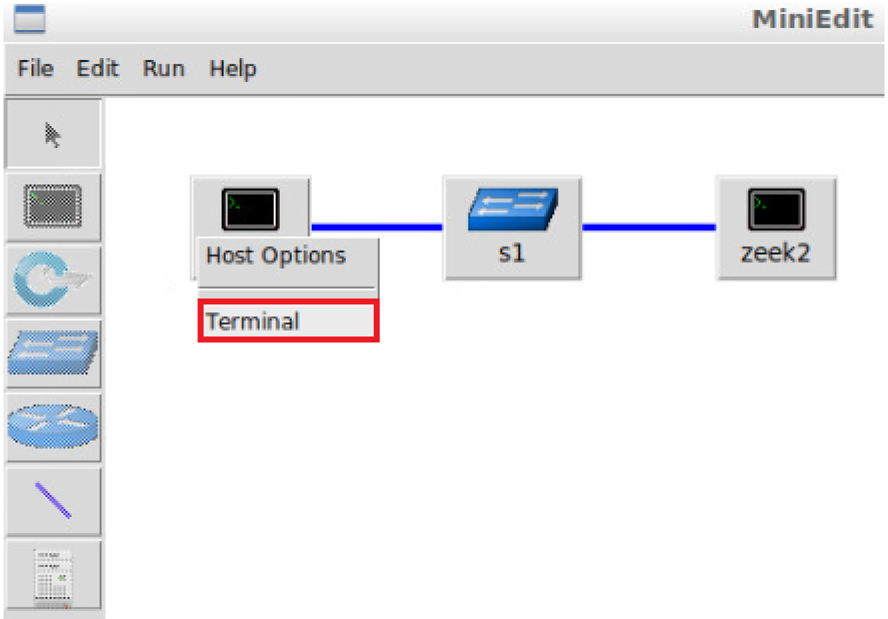


```
root@admin:~/Zeek-Labs/TCP-Traffic# tcpdump -i zeek2-eth0 -s 0 -w scantraffic.p
cap
tcpdump: listening on zeek2-eth0, link-type EN10MB (Ethernet), capture size 2621
44 bytes
```

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.

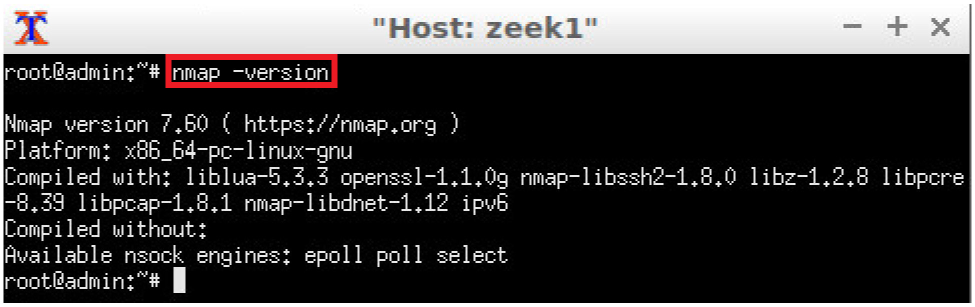
2.4 Using the zeek1 virtual machine for network scanning activities

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. On a machine running Linux, `nmap` is executed through the Terminal. Verify that `nmap` is functioning properly by viewing the currently installed version.

```
nmap -version
```



```
root@admin:~# nmap -version

Nmap version 7.60 ( https://nmap.org )
Platform: x86_64-pc-linux-gnu
Compiled with: liblua-5.3.3 openssl-1.1.0g nmap-libssh2-1.8.0 libz-1.2.8 libpcre
-8.39 libpcap-1.8.1 nmap-libdnet-1.12 ipv6
Compiled without:
Available nsock engines: epoll poll select
root@admin:~#
```

The figure above shows that the currently installed version of nmap is 7.60. With both the *zeek2* and *zeek1* virtual machines configured correctly, we can proceed with the exercises.

2.4.1 nmap options

`nmap` is used to discover hosts and services on a computer network by sending packets and analyzing the responses. `nmap` command has a list of options for every scan type and covers several protocols. This lab focuses on TCP scans with their default settings. Two additional options that can be used during this lab are:

- `-A`: enables operating system and version detection.

- `-T4`: faster execution, can strain the initiator's machine on larger scans.

More information is available on the following `nmap` documentation page:

```
https://nmap.org/book/man-briefoptions.html
```

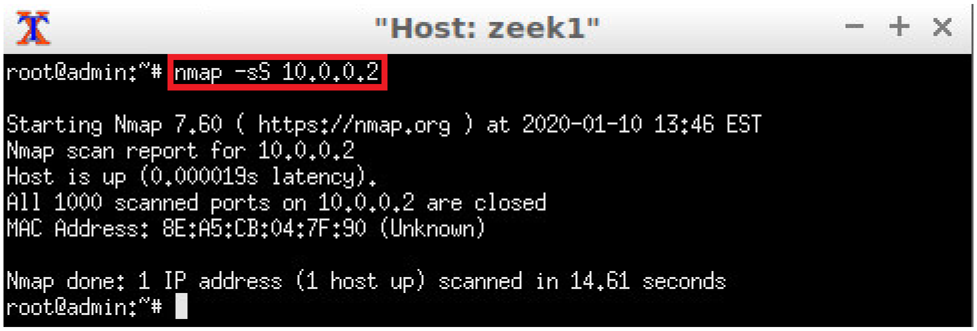
2.4.2 TCP SYN scans

TCP SYN scans are one of the most common types of scans used for vulnerability detection. During SYN scanning, the initiating host sends a single TCP SYN packet to the destination. The receiving host interprets the request as a new TCP connection where the standard three-way TCP handshake is to be established. If a SYN/ACK packet is sent back, the initiator can infer that the port is open. The initiator can then send an RST (reset) packet to terminate the established connection.

Step 1. Use the following command to conduct a TCP SYN scan.

```
nmap -sS 10.0.0.2
```

The `-sS` option is used to indicate a TCP SYN scan.



```
root@admin:~# nmap -sS 10.0.0.2

Starting Nmap 7.60 ( https://nmap.org ) at 2020-01-10 13:46 EST
Nmap scan report for 10.0.0.2
Host is up (0.000019s latency).
All 1000 scanned ports on 10.0.0.2 are closed
MAC Address: 8E:A5:CB:04:7F:90 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 14.61 seconds
root@admin:~#
```

After the scan is completed, `nmap` produces a report on the performed scan. This includes the scan starting time, the number of ports, the total time, etc. We can see here the TCP SYN scan took 14.61 seconds, and none of the scanned ports were open.

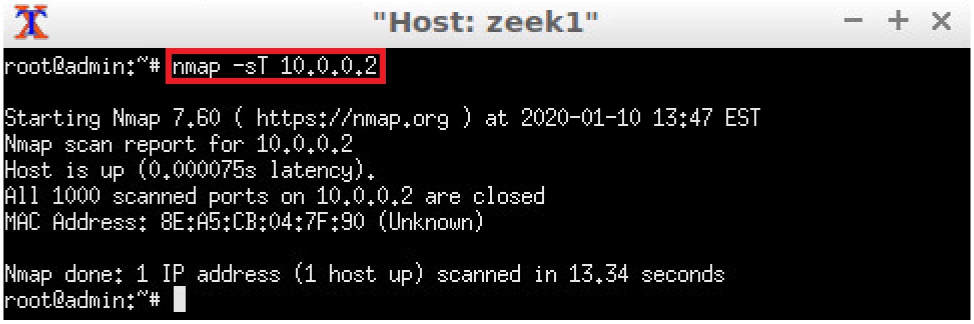
2.4.3 TCP connect scans

TCP Connect scans are an alternative to TCP SYN scans. Rather than starting a TCP handshake, the initiator's operating system attempts to establish a connection with the target victim through a system call. If a connection is successfully created, the initiator can infer that the receiver is open.

Step 1. Use the following command to conduct a TCP connect scan.

```
nmap -sT 10.0.0.2
```

The `-sT` option is used to indicate a TCP Connect scan.



```
root@admin:~# nmap -sT 10.0.0.2

Starting Nmap 7.60 ( https://nmap.org ) at 2020-01-10 13:47 EST
Nmap scan report for 10.0.0.2
Host is up (0.000075s latency).
All 1000 scanned ports on 10.0.0.2 are closed
MAC Address: 8E:A5:CB:04:7F:90 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 13.34 seconds
root@admin:~#
```

The report in the above figure shows that the scan was completed in 13.34 seconds, and none of the scanned ports were open.

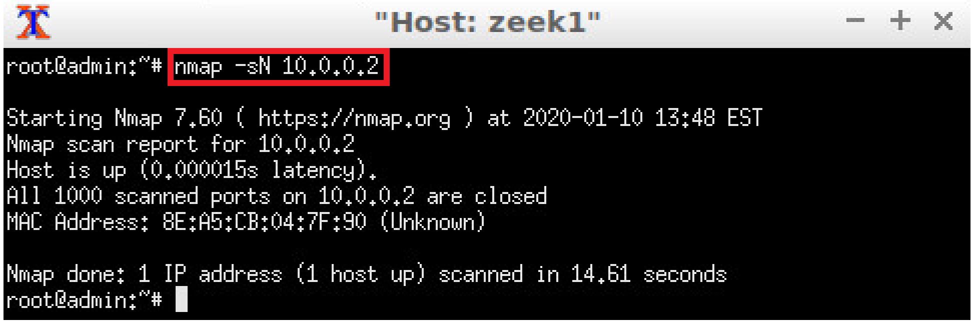
2.4.4 TCP NULL scans

TCP NULL scans are another form of TCP scanning. In general, all TCP packets contain flags. Firewalls are configured to drop packets containing certain flags. The TCP NULL scan attempts to bypass these firewalls by excluding the header. With a sequence number of 0, packets in a TCP NULL scan will have no flags and can potentially infiltrate a network's firewall.

Step 1. Use the following command to conduct a TCP NULL scan.

```
nmap -sN 10.0.0.2
```

The `-sN` option is used to indicate a TCP NULL scan.



```
root@admin:~# nmap -sN 10.0.0.2

Starting Nmap 7.60 ( https://nmap.org ) at 2020-01-10 13:48 EST
Nmap scan report for 10.0.0.2
Host is up (0.000015s latency).
All 1000 scanned ports on 10.0.0.2 are closed
MAC Address: 8E:A5:CB:04:7F:90 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 14.61 seconds
root@admin:~#
```

The report in the above figure shows that the scan was completed in 14.61 seconds, and none of the scanned ports were open.

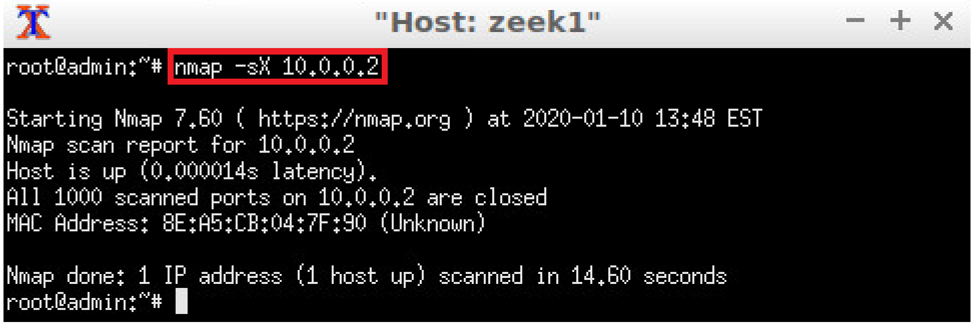
2.4.5 TCP XMAS scans

TCP Xmas scans, also known as Christmas tree scans, have their name derived from their set flags. In TCP Xmas scans, the PSH, URG and FIN flags are all set in the TCP header. This combination of flags is used in an attempt to infiltrate a strict network's firewall.

Step 1. Use the following command to conduct a TCP XMAS scan.

```
nmap -sX 10.0.0.2
```

The `-sX` option is used to indicate a TCP XMAS scan.

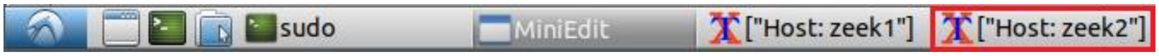


```
root@admin:~# nmap -sX 10.0.0.2

Starting Nmap 7.60 ( https://nmap.org ) at 2020-01-10 13:48 EST
Nmap scan report for 10.0.0.2
Host is up (0.000014s latency).
All 1000 scanned ports on 10.0.0.2 are closed
MAC Address: 8E:A5:CB:04:7F:90 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 14.60 seconds
root@admin:~#
```

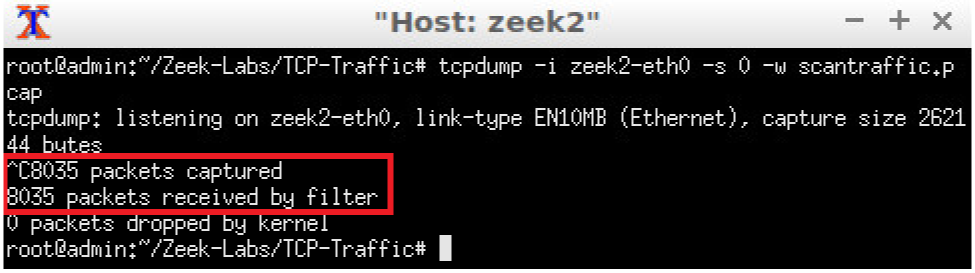
The report in the above figure shows that the scan was completed in 14.60 seconds, and none of the scanned ports were open or vulnerable.

2.4.6 Terminating live network capture

Step 1. 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.

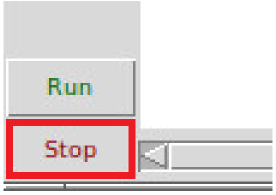


Step 2. Use the `Ctrl+c` key combination to stop live traffic capture. Statistics of the capture session will we be displayed. 8035 packets were recorded by the interface, which were then captured and stored in the new *scantraffic.pcap* file.



```
root@admin:~/Zeek-Labs/TCP-Traffic# tcpdump -i zeek2-eth0 -s 0 -w scantraffic.p
cap
tcpdump: listening on zeek2-eth0, link-type EN10MB (Ethernet), capture size 2621
44 bytes
^C8035 packets captured
8035 packets received by filter
0 packets dropped by kernel
root@admin:~/Zeek-Labs/TCP-Traffic#
```

Step 3. 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 `x` on the top right of the editor.

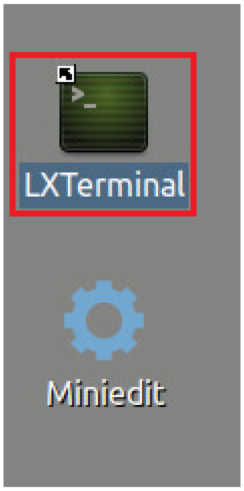


We will now return to the *Client* machine to process and analyze the newly generated network traffic.

3 Analyzing collected network traffic

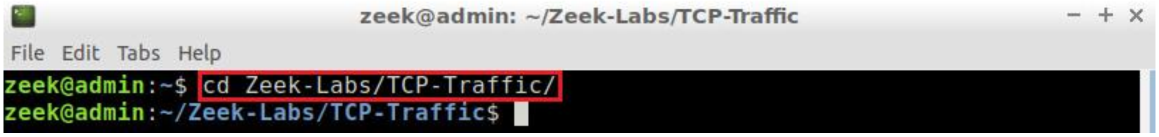
After successfully conducting a number of TCP-based scans, the *scanpackets.pcap* packet capture file now contains the required network traffic. In this section we analyze the collected network traffic using Zeek.

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 *scantraffic.pcap* file.

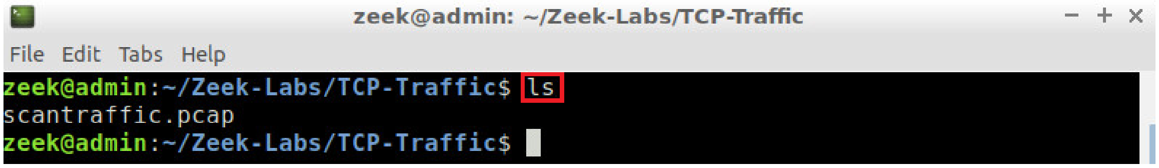
```
cd Zeek-Labs/TCP-Traffic/
```



```
zeek@admin:~$ cd Zeek-Labs/TCP-Traffic/
zeek@admin:~/Zeek-Labs/TCP-Traffic$
```

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

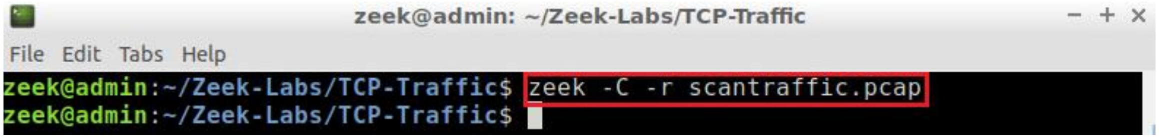
```
ls
```



```
zeek@admin:~/Zeek-Labs/TCP-Traffic$ ls
scantraffic.pcap
zeek@admin:~/Zeek-Labs/TCP-Traffic$
```

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

```
zeek -C -r scantraffic.pcap
```

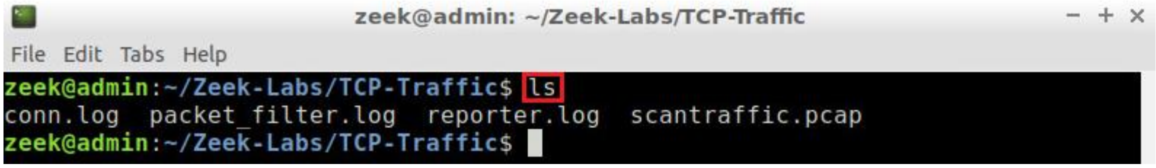


```
zeek@admin:~/Zeek-Labs/TCP-Traffic$ zeek -C -r scantraffic.pcap
zeek@admin:~/Zeek-Labs/TCP-Traffic$
```

Similarly to the previous labs, Zeek will process the *scantraffic.pcap* file and generate resulting log files based off of the default Zeek configurations.

Step 5. List the generated Zeek log files.

```
ls
```



```
zeek@admin:~/Zeek-Labs/TCP-Traffic$ ls
conn.log  packet_filter.log  reporter.log  scantraffic.pcap
zeek@admin:~/Zeek-Labs/TCP-Traffic$
```

With the log files generated, we can now use the `zeek-cut` utility for further analysis.

3.1 Example Query 1

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

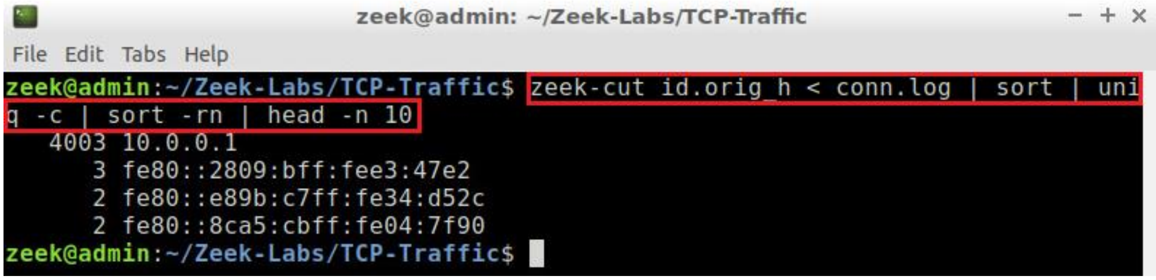
Step 1. Enter the following command.

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

The above command is explained as follows:

- `zeek-cut id.orig_h < conn.log`: selects the `id.orig_h` column from the *conn.log* file.
- `| sort`: uses the `sort` command to organize the rows in alphabetical order.
- `| uniq -c`: uses the `uniq` command with the `-c` option to remove duplicates while returning unique instances and their counts.

- `| sort -rn`: uses the `sort` command with the `-rn` option to organize the rows in reverse numerical order.
- `| head -n 10`: uses the `head` command with the `-n` option to display the 10 topmost values.



```
zeek@admin:~/Zeek-Labs/TCP-Traffic$ zeek-cut id.orig_h < conn.log | sort | uni
q -c | sort -rn | head -n 10
   4003 10.0.0.1
      3 fe80::2809:bff:fee3:47e2
      2 fe80::e89b:c7ff:fe34:d52c
      2 fe80::8ca5:cbff:fe04:7f90
zeek@admin:~/Zeek-Labs/TCP-Traffic$
```

We can see the majority of the packets were received from the *zeek1* machine denoted by the IP address 10.0.0.1.

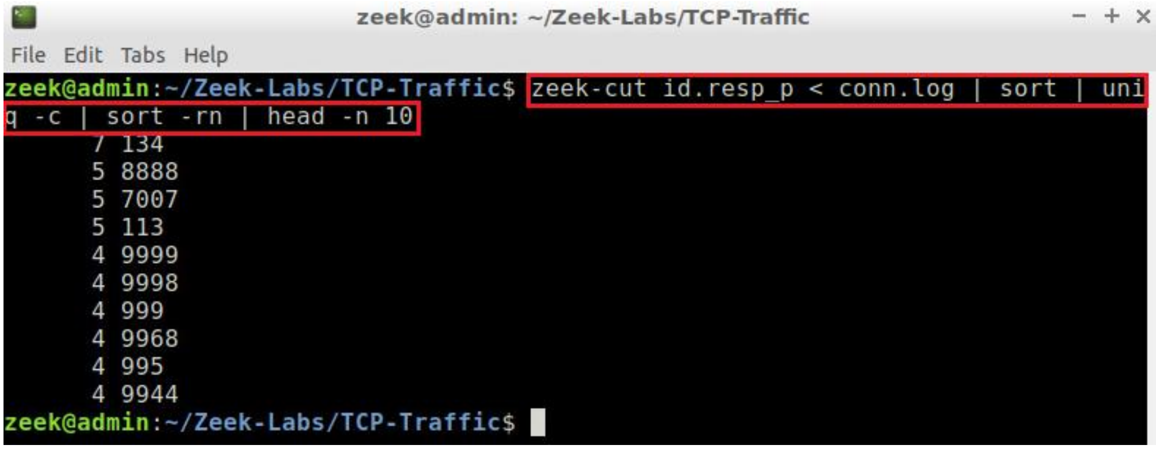
3.2 Example Query 2

Example 2: Show the 10 destination ports that received the most network traffic, organized in descending order.
Step 1. Enter the following command.

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

The above command is explained as follows:

- `zeek-cut id.orig_h < conn.log`: selects the `id.orig_h` column from the *conn.log* file.
- `| sort`: uses the `sort` command to organize the rows in alphabetical order.
- `| uniq -c`: uses the `uniq` command with the `-c` option to remove duplicates while returning unique instances and their counts.
- `| sort -rn`: uses the `sort` command with the `-rn` option to organize the rows in reverse numerical order.
- `| head -n 10`: uses the `head` command with the `-n` option to display the 10 topmost values.



```
zeek@admin:~/Zeek-Labs/TCP-Traffic$ zeek-cut id.resp_p < conn.log | sort | uni
q -c | sort -rn | head -n 10
      7 134
      5 8888
      5 7007
      5 113
      4 9999
      4 9998
      4 999
      4 9968
      4 995
      4 9944
zeek@admin:~/Zeek-Labs/TCP-Traffic$
```

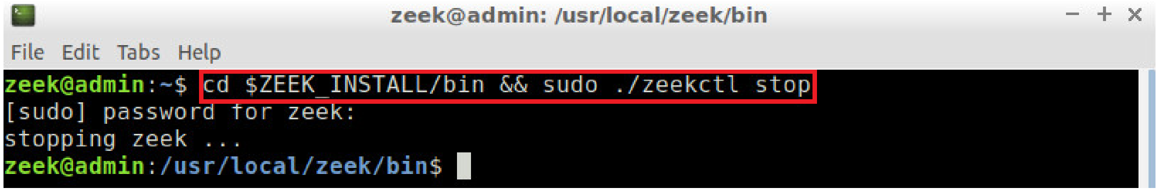
The number of duplicates is seen in the left column, while the matching destination port is seen in the right column. More than 10 unique destination ports were found, so only the top 10 were returned. These destination ports may be variable due to `nmap’s` scanning configurations.

3.3 Closing the current instance of Zeek

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

Concluding this lab, we have reviewed the steps required to generate scan traffic as well as enable live traffic capture using Zeek. Once collected, the trace files can be studied, and empirical data can be investigated regarding the current state of a network and its devices.

References

1. Bou-Harb, Elias, Mourad Debbabi, and Chadi Assi. "A systematic approach for detecting and clustering distributed cyber scanning." Computer Networks 57.18 (2013): 3826-3839.
2. Pour, Morteza Safaei, and Elias Bou-Harb. "Implications of theoretic derivations on empirical passive measurements for effective cyber threat intelligence generation." *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018.
3. “Options summary”, nmap, [Online], Available: nmap, https://nmap.org/book/man-briefoptions.html.
4. “Port scanning techniques”, nmap, [Online], Available: nmap, https://nmap.org/book/man-port-scanning-techniques.html.