CAPTURE YOUR FIRST PACKET WITH TCPDUMP

Pedro Osorio Lopez.

1. High-Level Project Description:

*In this activity, we'll capture and analyze live network traffic using tcpdump. We'll use Linux commands in the Bash shell to perform these tasks.

2. Project Setup and Required Tools:

*A Linux Machine with tcpdump is required. Note that the setup was already provided to us in the course, so please refer to the appropriate guides on the internet if necessary.

3. Step by Step Project Walkthrough

*In this activity we will: Identify available network interfaces, use tcpdump to capture live network traffic, save network traffic to a packet capture file, and filter the packet capture data.

*We'll start by running the tcpdump command to automatically capture network traffic. Note that sudo permissions might be necessary, and the number for packets can be specified with the flag -c.

```
┌──(kali㉿kali)-[~]
└─$ sudo tcpdump
[sudo] password for kali:
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
15:32:01.352151 IP 10.0.2.15.47447 > homerouter.cpe.domain: 6798+ A? contile.services.mozilla.com. (46)
15:32:01.352383 IP 10.0.2.15.47447 > homerouter.cpe.domain: 25331+ AAAA? contile.services.mozilla.com. (46)
15:32:01.407168 IP homerouter.cpe.domain > 10.0.2.15.47447: 6798 1/0/0 A 34.117.237.239 (62)
15:32:01.418501 IP homerouter.cpe.domain > 10.0.2.15.47447: 25331 0/1/0 (127)
15:32:01.419545 IP 10.0.2.15.46290 > 239.237.117.34.bc.googleusercontent.com.https: Flags [S], seq 4293174533, win
64240, options [mss 1460,sackOK,TS val 1027448125 ecr 0,nop,wscale 7], length 0
15:32:01.477014 IP 239.237.117.34.bc.googleusercontent.com.https > 10.0.2.15.46290: Flags [S.], seq 34752001, ack 4
293174534, win 65535, options [mss 1460], length 0
15:32:01.477037 IP 10.0.2.15.46290 > 239.237.117.34.bc.googleusercontent.com.https: Flags [.], ack 1, win 64240, le
ngth 0
15:32:01.581160 IP 10.0.2.15.46290 > 239.237.117.34.bc.googleusercontent.com.https: Flags [P.], seq 1:518, ack 1, w
in 64240, length 517
15:32:01.581515 IP 239.237.117.34.bc.googleusercontent.com.https > 10.0.2.15.46290: Flags [.], ack 518, win 65535,
length 0
15:32:01.629218 IP 10.0.2.15.46803 > homerouter.cpe.domain: 65080+ PTR? 1.8.168.192.in-addr.arpa. (42)
15:32:01.637412 IP 239.237.117.34.bc.googleusercontent.com.https > 10.0.2.15.46290: Flags [P.], seq 1:2821, ack 518
, win 65535, length 2820
15:32:01.637431 IP 10.0.2.15.46290 > 239.237.117.34.bc.googleusercontent.com.https: Flags [.], ack 2821, win 62780,
 length 0
15:32:01.647023 IP 239.237.117.34.bc.googleusercontent.com.https > 10.0.2.15.46290: Flags [P.], seq 2821:4517, ack
518, win 65535, length 1696
15:32:01.647037 IP 10.0.2.15.46290 > 239.237.117.34.bc.googleusercontent.com.https: Flags [.], ack 4517, win 62780,
 length 0
```

*After the capture is completed (press control+c if needed), we will proceed to save the capture using the command sudo tcpdump -i eth0 -w capture.pcap. We can then view this capture with the command sudo tcpdump -r capture.pcap.

```
┌──(kali㉿kali)-[~]
└─$ sudo tcpdump -i eth0 -w capture.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C19 packets captured
19 packets received by filter
0 packets dropped by kernel
```

```
┌──(kali㉿kali)-[~]
└─$ sudo tcpdump -r capture.pcap
[sudo] password for kali:
reading from file capture.pcap, link-type EN10MB (Ethernet), snapshot length 262144
15:47:28.549552 IP 10.0.2.15.56088 > mad07s23-in-f3.1e100.net.http: Flags [.], ack 136772211, win 63882, length 0
15:47:28.550881 IP mad07s23-in-f3.1e100.net.http > 10.0.2.15.56088: Flags [.], ack 1, win 65535, length 0
15:47:28.804709 IP 10.0.2.15.56082 > mad07s23-in-f3.1e100.net.http: Flags [.], ack 136261615, win 63882, length 0
15:47:28.805057 IP mad07s23-in-f3.1e100.net.http > 10.0.2.15.56082: Flags [.], ack 1, win 65535, length 0
15:47:29.060686 IP 10.0.2.15.35322 > server-18-154-40-210.mad53.r.cloudfront.net.http: Flags [.], ack 144320946, wi
n 63296, length 0
15:47:29.061115 IP server-18-154-40-210.mad53.r.cloudfront.net.http > 10.0.2.15.35322: Flags [.], ack 1, win 65535,
 length 0
```

*Lastly, we can use commands such as sudo tcpdump -i eth0 dst port 80 to capture traffic for specific ports only (in this case port 80, HTTP).

```
┌──(kali㉿kali)-[~]
└─$ sudo tcpdump -i eth0 dst port 80
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
16:54:35.688739 IP 10.0.2.15.45486 > mad41s08-in-f3.1e100.net.http: Flags [.], ack 639042105, win 63791, length 0
16:54:38.504838 IP 10.0.2.15.45472 > mad41s08-in-f3.1e100.net.http: Flags [.], ack 638978806, win 63791, length 0
16:54:45.925317 IP 10.0.2.15.45486 > mad41s08-in-f3.1e100.net.http: Flags [.], ack 1, win 63791, length 0
16:54:48.747473 IP 10.0.2.15.45472 > mad41s08-in-f3.1e100.net.http: Flags [.], ack 1, win 63791, length 0
16:54:56.168790 IP 10.0.2.15.45486 > mad41s08-in-f3.1e100.net.http: Flags [.], ack 1, win 63791, length 0
16:54:58.980712 IP 10.0.2.15.45472 > mad41s08-in-f3.1e100.net.http: Flags [.], ack 1, win 63791, length 0
^C
6 packets captured
7 packets received by filter
0 packets dropped by kernel
```

4. Summary and/or Recommendations:

*In this activity, we learned about a powerful tool called tcpdump that can be used to capture and analyze network traffic..