OLD DOMINION UNIVERSITY

CYSE 301: CYBERSECURITY TECHNIQUES AND OPERATIONS

FALL 2017

Module I
Traffic Tracing and Analysis

Topic 5 Use tcpdump to Capture Network Traffic

1. Introduction

In this module, we are going to learn about the basic network structures and the way of simple network defense and countermeasures. As a network administrator, if we want to set up and maintain a simple functionality and security network, we are not only need to master the fundamental knowledge of the network but also need to operate and configure the network facility such as the switch, router and firewall in the field and track the trace of the package through the network to deeply understand how to do cyber defense from the very beginning. Also, as a network administrator, one essential ability is to capture and analyze network traffic. This can be important to identify the cause of the bottleneck, determining who is responsible for the certain intrusion.

2. Objective

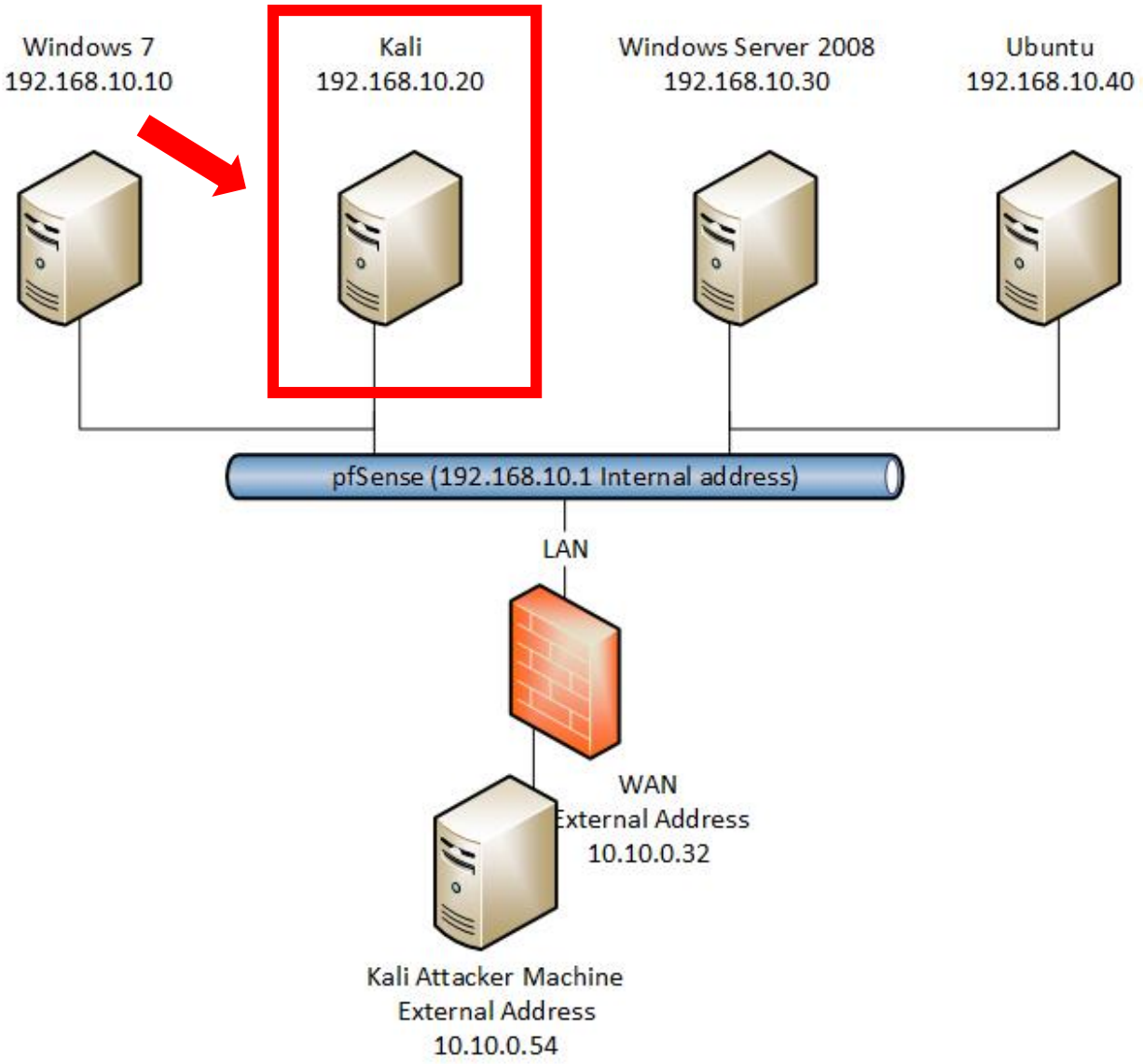
The objective of this topic is to master the command line version traffic sniffer, tcpdump, to capture and display the packets when an interactive user interface isn't necessary or available. It allows the user to display TCP/IP and other packets being transmitted or received over a network to which the computer is attached.

WARNING:

Listening, sniffing, eavesdropping on networks to which you do not have legal access is unethical and may even constitute a crime in your area.

NETWORK TOPOLOGY

In this lab, you need to login Kali VM to complete all the assignments.



3. GETTING STARTED WITH TCPDUMP

3.1 What is tcpdump

tcpdump is an industry-standard packet sniffer. And it is a most powerful and widely used command-line packets sniffer or package analyzer tool which is used to capture or filter TCP/IP packets that received or transferred over a network on a specific interface. It is available in most of the Linux/Unix based operating systems. tcpdump also gives us an option to save captured packets to a file for future analysis. It saves the file in a pcap format, that can be viewed by tcpdump command or an open source GUI based tool called Wireshark (Network Protocol Analyzer) that reads tcpdump pcap format files

When using a tool that displays network traffic a more natural (raw) way the burden of analysis is placed directly on the human rather than the application (Figure 1).

```
15:31:34.079416 IP (tos 0x0, ttl 64, id 20244, offset 0, flags [DF],
proto: TCP (6), length: 60) source.35970 > dest.80: S, cksum 0x0ac1
(correct), 2647022145:2647022145(0) win 5840 0x0000: 4500 003c 4f14
4006 7417 0afb 0257  E..  0x0010: 4815 222a 8c82 0050 9dc6 5a41 0000
0000  H."*...P..ZA.... 0x0020: a002 16d0 0ac1 0000 0204 05b4
0402 080a  ..............  0x0030: 14b4 1555 0000 0000 0103 0302
```

Figure 1 Raw TCP/IP output

In the following sections, we are going to introduce some basic operations in tcpdump.

3.2 Basic operations

Here list the basic command line options while using tcpdump.

-A	Print frame payload in ASCII	**-q**	Quick output
-c <count>	Exit after capturing **count** packets	**-r <file>**	Read packets from **file**
-D	List available interfaces	**-s <len>**	Capture up to **len** bytes per packet
-e	Print link-level headers	**-S**	Print absolute TCP sequence numbers
-F <file>	Use **file** as the filter expression	**-t**	Don't print timestamps
-G <n>	Rotate the dump file every n seconds	**-v[v[v]]**	Print more verbose output
-i <iface>	Specifies the capture interface	**-w <file>**	Write captured packets to **file**
-K	Don't verify TCP checksums	**-x**	Print frame payload in hex
-L	List data link types for the interface	**-X**	Print frame payload in hex and ASCII
-n	Don't convert addresses to names	**-y <type>**	Specify the data link type
-p	Don't capture in promiscuous mode	**-Z <user>**	Drop privileges from root to **user**

Figure 2

3.2.1. Display Available Interfaces

To list number of available interfaces on the system, run the following command with ***-D*** option

#tcpdump -D

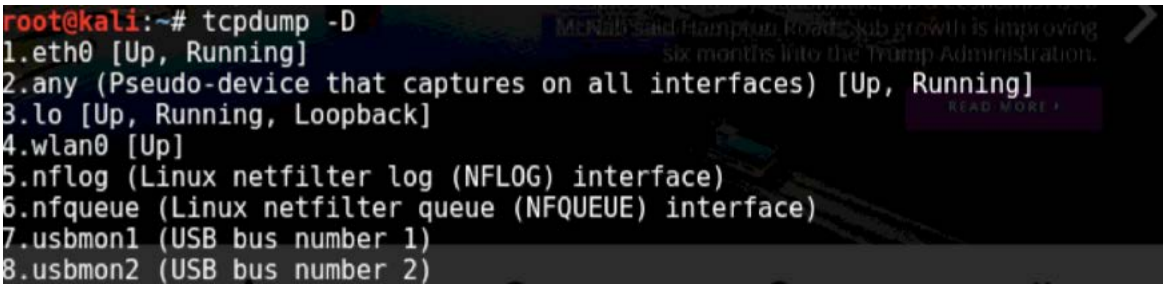


Figure 3 Show available interfaces

3.2.2. Capture Packets from Specific Interface

The command screen will scroll up until you interrupt and when we execute tcpdump command it will capture from all the interfaces, however with ***-i*** switch only capture from desired interface.

#tcpdump –i eth0

```
root@kali:~# tcpdump -i eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
11:42:39.467960 ARP, Request who-has gateway tell kali, length 28
11:42:39.468479 ARP, Reply gateway is-at 00:50:56:f1:59:24 (oui Unknown), length 46
11:42:39.469880 IP kali.39138 > gateway.domain: 34636+ PTR? 2.174.168.192.in-addr.arpa. (44)
11:42:39.471869 IP gateway.domain > kali.39138: 34636 NXDomain 0/1/0 (106)
11:42:39.473617 IP kali.59168 > gateway.domain: 43625+ PTR? 129.174.168.192.in-addr.arpa. (46)
11:42:39.475428 IP gateway.domain > kali.59168: 43625 NXDomain 0/1/0 (108)
```

Figure 4 tcpdump choose network interface

3.2.3. Capture Only N Number of Packets

When you run tcpdump command, it will capture all the packets for specified interface, until you Hit cancel button. But using ***-c*** option, you can capture specified number of packets. The below example will only capture 10 packets.

#tcpdump –i eth0 –c 10

```
root@kali:~# tcpdump -i eth0 -c 10
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
11:45:38.028076 IP kali.33606 > ec2-50-112-201-212.us-west-2.compute.amazonaws.com.https: Flags [.]
, ack 88438966, win 37960, length 0
11:45:38.028466 IP ec2-50-112-201-212.us-west-2.compute.amazonaws.com.https > kali.33606: Flags [.]
, ack 1, win 64240, length 0
11:45:38.028525 IP kali.39564 > gateway.domain: 29823+ PTR? 212.201.112.50.in-addr.arpa. (45)
11:45:38.051972 IP gateway.domain > kali.39564: 29823 1/0/0 PTR ec2-50-112-201-212.us-west-2.comput
e.amazonaws.com. (109)
11:45:38.052097 IP kali.47342 > gateway.domain: 59874+ PTR? 129.174.168.192.in-addr.arpa. (46)
11:45:38.053096 IP gateway.domain > kali.47342: 59874 NXDomain 0/1/0 (108)
11:45:38.053386 IP kali.39283 > gateway.domain: 44688+ PTR? 2.174.168.192.in-addr.arpa. (44)
11:45:38.054212 IP gateway.domain > kali.39283: 44688 NXDomain 0/1/0 (106)
11:45:41.801426 IP kali.37772 > gateway.domain: 19004+ A? www.odu.edu. (29)
11:45:41.801480 IP kali.37772 > gateway.domain: 60017+ AAAA? www.odu.edu. (29)
10 packets captured
10 packets received by filter
0 packets dropped by kernel
```

Figure 5 Capture only N number of packets

3.2.4. Print Captured Packets in ASCII

The below tcpdump command with option -A displays the package in ASCII format. It is a character-encoding scheme format.

#tcpdump –A –i eth0

This part is intentionally left blank. Please screenshot the result and attach it in your lab report.

3.2.5. Display Captured Packets in HEX and ASCII

The following command with option ***-XX*** capture the data of each packet, including its link level header in HEX and ASCII format.

#tcpdump –XX -I eth0

This part is intentionally left blank. Please screenshot the result and attach it in your lab report.

3.2.6. Capture and Save Packets in a File

As we said, that tcpdump has a feature to capture and save the file in a. pcap format, to do this just execute the command with ***-w*** option.

 # tcpdump -w test.pcap -i eth0

3.3 Capture Filter Primitives

Here list the basic capture filters while using tcpdump.

`[src|dst] host <host>`	Matches a host as the IP source, destination, or either
`ether [src|dst] host <ehost>`	Matches a host as the Ethernet source, destination, or either
`gateway host <host>`	Matches packets which used **host** as a gateway
`[src|dst] net <network>/<len>`	Matches packets to or from an endpoint residing in **network**
`[tcp|udp] [src|dst] port <port>`	Matches TCP or UDP packets sent to/from **port**
`[tcp|udp] [src|dst] portrange <p1>-<p2>`	Matches TCP or UDP packets to/from a port in the given range
`less <length>`	Matches packets less than or equal to **length**
`greater <length>`	Matches packets greater than or equal to **length**
`(ether|ip|ip6) proto <protocol>`	Matches an Ethernet, IPv4, or IPv6 protocol
`(ether|ip) broadcast`	Matches Ethernet or IPv4 broadcasts
`(ether|ip|ip6) multicast`	Matches Ethernet, IPv4, or IPv6 multicasts
`type (mgt|ctl|data) [subtype <subtype>]`	Matches 802.11 frames based on type and optional subtype
`vlan [<vlan>]`	Matches 802.1Q frames, optionally with a VLAN ID of **vlan**
`mpls [<label>]`	Matches MPLS packets, optionally with a label of **label**
`<expr> <relop> <expr>`	Matches packets by an arbitrary expression

Figure 6

3.3.1. Match packets from source and destination IP

To capture packets from source or destination IP, just use command line parameter **dst** or **src**. Take one of ODU's IP address for example

 # tcpdump -i eth0 src 128.82.112.29

 # tcpdump -i eth0 det 192.168.2.15

This part is intentionally left blank. Please screenshot the result and attach it in your lab report.

3.3.2. Match packets from Specific Port

Let's say you want to capture packets for specific port 80 to fetch HTTP traffic steam, execute the below command by specifying port number 80 as shown below.

 # tcpdump -i eth0 port 80 -w test.pcap

This part is intentionally left blank. Please screenshot the result and attach it in your lab report.

3.3.3. Match packets for a specific protocol

To capture packets based on different protocols, run the following command with option **<protocol>.** For example

 # tcpdump -i eth0 tcp

This part is intentionally left blank. Please screenshot the result and attach it in your lab report.

#tcpdump -i eth0 icmp

This part is intentionally left blank. Please screenshot the result and attach it in your lab report.

Here attached the list of protocols filter (Figure 7) can be applied while using tcpdump and different ICMP types

arp	ip6	slip
ether	link	tcp
fddi	ppp	tr
icmp	radio	udp
ip	rarp	wlan

Figure 7 Protocols

ICMP Types		
icmp-echoreply	icmp-routeradvert	icmp-tstampreply
icmp-unreach	icmp-routersolicit	icmp-ireq
icmp-sourcequench	icmp-timxceed	icmp-ireqreply
icmp-redirect	icmp-paramprob	icmp-maskreq
icmp-echo	icmp-tstamp	icmp-maskreply

Figure 8 ICMP types

Moreover, we can use *modifiers* to customize the filter. Figure 9 shows three modifiers, and the usage example.

Modifiers	Examples	
! or not	udp dst port not 53	UDP not bound for port 53
&& or and	host 10.0.0.1 && host 10.0.0.2	Traffic between these hosts
|| or or	tcp dst port 80 or 8080	Packets to either TCP port

Figure 9 Modifiers and examples

.