Ashlyn Cooper
CPSC 3600
Assignment 3

Part II – Explanation of the tcpdump tool

Tcpdump is a command line utility that is a network capture and protocol analysis tool. It allows the user to analyze network traffic going through their system. It helps with network troubleshooting and security. Tcpdump uses packet capture. It can capture packets based on a specified pattern or across a certain time. Tcpdump can allow you to display packets being transmitted or received over a network to which the user's computer is attached. Tcpdump can also be used to print the contents of network packets and it can read packets from a network interface card. It can also write packets to standard output or a file. Essentially, tcpdump acts as a network sniffer to monitor traffic through a network.

I ran tcpdump in 2 ways.

For the first way I ran tcpdump with flags that would have it capture all packets in any interface. It captured packets until I ran the command Ctrl+C to interrupt the process. In this case it captured 7 packets until I interrupted the process. As it captures the packets, it shows me information about the packets and automatically resolves IP addresses to domain names. The information displayed can differ based on the protocol. In this case the first piece of info in the line is a time stamp of the received packet according to the local clock. It also then displays the protocol used like IP6 or IP. The it displays the source IP address and port and then the destination IP address and port. Then it displays the TCP flags

```
cpsc3600@vm1-ubuntu-1804:~$ sudo tcpdump --interface any
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked), capture size 262144 bytes
16:29:39.523370 IP6 vm1-ubuntu-1804.mdns > ff02::fb.mdns: 0 PTR (QM)? _pgpkey-hk
p._tcp.local. (40)
16:29:39.523527 IP vm1-ubuntu-1804.mdns > 224.0.0.251.mdns: 0 PTR (QM)? _pgpkey-
hkp._tcp.local. (40)
16:29:39.523577 IP localhost.mdns > 224.0.0.251.mdns: 0 PTR (QM)? _pgpkey-hkp._t
cp.local. (40)
16:29:39.524730 IP localhost.33797 > localhost.domain: 59863+ [1au] PTR? b.f.0.0
.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.f.f.ip6.arpa. (101)
16:29:39.525541 IP vm1-ubuntu-1804.36749 > dns.google.domain: 3399+ [1au] PTR? b
.f.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.f.f.ip6.arpa. (101)
16:29:39.605751 IP localhost.39095 > localhost.domain: 20756+ [1au] PTR? 53.0.0.
127.in-addr.arpa. (52)
16:29:39.605868 IP localhost.domain > localhost.39095: 20756 1/0/1 PTR localhost
. (75)
^C
7 packets captured
44 packets received by filter
31 packets dropped by kernel
```

The second way I ran tcpdump was by using options in the command that only captures 5 packets that use ICMP as the protocol. In another terminal window I pinged another machine and in the tcpdump capture shown below the tool only captured and displayed 5 ICMP related packets because those were the options I used in the command.

```
cpsc3600@vm1-ubuntu-1804:~$ sudo tcpdump -i any -c5 icmp
[sudo] password for cpsc3600:
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked), capture size 262144 bytes
16:31:43.653971 IP vm1-ubuntu-1804 > ec2-54-204-39-132.compute-1.amazonaws.com:
ICMP echo request, id 2889, seq 1, length 64
16:31:43.677381 IP ec2-54-204-39-132.compute-1.amazonaws.com > vm1-ubuntu-1804:
ICMP echo reply, id 2889, seq 1, length 64
16:31:44.656296 IP vm1-ubuntu-1804 > ec2-54-204-39-132.compute-1.amazonaws.com:
ICMP echo request, id 2889, seq 2, length 64
16:31:44.678926 IP ec2-54-204-39-132.compute-1.amazonaws.com > vm1-ubuntu-1804:
ICMP echo reply, id 2889, seq 2, length 64
16:31:45.657563 IP vm1-ubuntu-1804 > ec2-54-204-39-132.compute-1.amazonaws.com:
ICMP echo request, id 2889, seq 3, length 64
5 packets captured
5 packets received by filter
0 packets dropped by kernel
```

Extra Credit

Netcat is a utility program that helps us manage networks and monitor traffic between systems that was released in 1995. Netcat can be used to help identify how a network is performing and what type of activity is happening on the network. The use of netcat can allows a user to scan ports and determine if one is open or not. Netcat is used for reading and writing to network connections using TCP or UDP. Netcat includes features such as port scanning, transferring files, port listening, and can be used as a backdoor.