Lab 8.a) Analyzing UDP datagrams using Wireshark:

Start your web browser and clear the browser's cache memory, but do not access any

website yet.

Open Wireshark and start capturing.

Go back to your web browser and retrieve any file from a website. Wireshark starts

capturing packets.

After enough packets have been captured, stop Wireshark and save the captured file.

Using the captured file, analyze TCP & UDP packets captured

Note: DNS uses UDP for name resolution , HTTP uses TCP.

Using the captured information, answer the following questions in your lab report.

A. In the packet list pane, select the first DNS packet. In the packet detail pane, select the

User Datagram Protocol. The UDP hexdump will be highlighted in the packet byte lane.

Using the hexdump, Answer the following:

a. the source port number.

b. the destination port number.

c. the total length of the user datagram.

d. the length of the data.

e. whether the packet is directed from a client to a server or vice versa.

f. the application-layer protocol.

g. whether a checksum is calculated for this packet or not.

b) Network Data Analysis using tcpdump

Tcpdump:

Tcpdump is a network traffic sniffer built on the packet capture library libpcap. While started, it

captures and displays packets on the LAN segment. By analyzing the traffic flows and the packet

header fields, a great deal of information can be gained about the behavior of the protocols and

their operation within the network. Problems in the network can also be identified. A packet filter

can be defined in the command line with different options to obtain a desired output.

Basics

Below are a few options you can use when configuring tcpdump.

Options

-i any : Listen on all interfaces just to see if you’re seeing any traffic.

-I eth0 : Listen on the eth0 interface.

-D : Show the list of available interfaces

-n : Don’t resolve hostnames.

-nn : Don’t resolve hostnames or port names.

-q : Be less verbose (more quiet) with your output.

-t : Give human-readable timestamp output.

-tttt : Give maximally human-readable timestamp output.

-X : Show the packet’s contents in both hex and ascii.

-XX : Same as -X, but also shows the ethernet header.

-c : Only get x number of packets and then stop.

-s : Define the snaplength (size) of the capture in bytes. Use -s0 to get everything, unless

you are intentionally capturing less.

-S : Print absolute sequence numbers.

-e : Get the ethernet header as well.

-q : Show less protocol information.

-E : Decrypt IPSEC traffic by providing an encryption key.

By looking at all interfaces.

# tcpdump -i any

SPECIFIC INTERFACE

# tcpdump -i eth0

# tcpdump -ttttnnvvS

FIND TRAFFIC BY IP

One of the most common queries, this will show you traffic from 1.2.3.4, whether it’s the source

or the destination.

# tcpdump host 1.2.3.4

FILTERING BY SOURCE AND DESTINATION

# tcpdump src 2.3.4.5

# tcpdump dst 3.4.5.6

FINDING PACKETS BY NETWORK

To find packets going to or from a particular network, use the netoption. You can combine this

with the src or dst options as well.

# tcpdump net 1.2.3.0/24

SHOW TRAFFIC RELATED TO A SPECIFIC PORT

You can find specific port traffic by using the port option followed by the port number.

# tcpdump port 3389

# tcpdump src port 1025

SHOW TRAFFIC OF ONE PROTOCOL

# tcpdump icmp

SHOW ONLY IP6 TRAFFIC

# tcpdump ip6

FIND TRAFFIC USING PORT RANGES

You can also use a range of ports to fin traffic.

# tcpdump portrange 21-23

FIND TRAFFIC BASED ON PACKET SIZE

# tcpdump less 32

# tcpdump greater 64

# tcpdump <= 128

Writing pcap file

# tcpdump port 80 -w capture\_file

reading pcap file

# tcpdump port 80 -w capture\_file