\input{header}

\begin{document}

\begin{center}

{\LARGE Review of Packet Capture Introspection}

\vspace{0.1in}\\

\end{center}

\section{Overview}

The pcap (packet capture) format is a standard and portable representation of packet-level

network traffic. You are likely already familiar with pcap – both Wireshark and tcpdump

store and read data in pcap format.

This introductory lab is designed to familiarize students with pcaps and traffic analysis

using Wireshark. Wireshark includes many powerful tools and is best suited to performing

highly targeted analysis on small packet captures. This lab is adapted from [1], which is a

helpful resource for improving your familiarity with the Wireshark toolset.

\subsection {Background}

The student is expected to have at least a basic understanding of the Linux command line,

the basics of the file system.

\section{Lab Environment}

This lab runs in the Labtainer framework,

available at http://my.nps.edu/web/c3o/labtainers.

That site includes links to a pre-built virtual machine

that has Labtainers installed, however Labtainers can

be run on any Linux host that supports Docker containers.

From your labtainer-student directory start the lab using:

\begin{verbatim}

labtainer packet-introspection

\end{verbatim}

\noindent A link to this lab manual will be displayed.

\section {Tasks}

\subsection {Find Most Active TCP Flow (15 pts)}

A common network analysis task is determining the largest contributors to network traffic

and potential congestion. In this part you will isolate and examine the largest TCP flow1

in

a packet capture. Complete the following steps and answer the questions.

\begin{itemize}

\item Open {\tt pcaps/http-misctraffic101.pcapng} in Wireshark

\item Select Statistics | Conversations. Click the Ethernet tab; notice there is only one

pair of hosts communicating on the local network. Ensure that the “Name resolution”

box is checked. The MAC address listed as “Cadant” is the local router. The

“Flextron” host is the client from which the traffic was captured.

\item[1.] [5 pts] Click on the IPv4 tab to examine the IPv4 conversations in this trace file. Based

on the bytes count, what IP addresses participate in the most active IPv4 conversation?

\item Click the TCP tab to identify the most active TCP conversation. Sort by bytes by

clicking on the Bytes column heading.

\item When looking at the most active flow, we see that host 107.6.133.250 is using port

“http” (80) and host 25.6.181.160 is using port “dellpwrappks” (1266). Since HTTP

clients choose a random ephemeral port to communicate, we can be reasonably confident that this traffic is in fact unrelated to the dellpwrappks service. (If you see service

names, you can uncheck the Name resolution box to view the port numbers.)

\item[2.] [10 pts] Right-click on the most active TCP conversation and select Apply as a Filter

| Selected | A ↔ B. Wireshark automatically creates and applies a display filter for

this TCP conversation. How many packets match this filter?

\item Part 1 clean-up: Click the Clear button (the red ’X’ next to the filter expression) to

remove your display filter before continuing. Toggle to the Conversation window and

click Close.

\end {itemize}

\subsection{Geolocating IP Addresses (15 pts)}

Correlating network interfaces/IP addresses to physical locations is often a useful task. Wireshark

includes a basic capability in this regard, which utilizes the free versions of the MaxMind2

database. It is important to recognize that no IP-geolocation database is error-free.

Later on in the quarter we will discuss various approaches to geolocating IP addresses and

the associated complexities of this process.

\begin{itemize}

\item Open {\tt pcaps/http-browse101c.pcapng} in Wireshark.

\item Now select Edit | Preferences

| Name Resolution4 and click the GeoIP database directories Edit button. Click

New and point to the maxmind directory (which has database files downloaded from

\url{http://dev.maxmind.com/geoip/legacy/geolite/)}. Continue to click OK until you have

closed the GeoIP database paths windows and the Preferences window.

\item Select Statistics | Endpoints and click on the IPv4 tab. You should see information

in the Country, City, Latitude, and Longitude columns.

\item Click the Map button. Wireshark will launch a map view in your browser with the

known IP addresses plotted as points on the map. Click on any of the plot point to

find more information about the IP address.

\item[3.] [15 pts] How much aggregate traffic went to/from Milpitas, CA?

\item Part 2 clean-up: Close the browser tab/window when you are finished. Close the

Wireshark Endpoints window

\end{itemize}

\subsection{Reassemble text from TCP stream (15 pts)}

As a byte-stream oriented protocol, TCP segments data based on its MSS, not based on

semantics of the English language, or even application data formatting. Thus it can be

helpful to reassemble this data before manually inspecting it.

\begin{itemize}

\item Open {\tt pcaps/http-wiresharkdownload101.pcapng} in Wireshark.

\item The first three packets are the TCP handshake for the web server connection. Frame

4 contains the client’s GET request for the download.html page. Right-click on frame

4 and select Follow | TCP stream. Traffic from the first host seen in the trace file,

the client in this case, is colored red. Traffic from the second host seen in the trace

file, the server in this case, is colored blue.

\item[4.] [5 points] Wireshark displays the conversation without the Ethernet, IP or TCP headers.

Scroll through the stream to look for the hidden message from Gerald Combs,

creator of Wireshark. It is located in the server stream and begins with “X-Slogan”.

What is the message?

\item This isn't the only message hidden in the web browsing session. Now that you know

the message begins with “X-Slogan”, you can have Wireshark display every frame that

includes this ASCII string. Click the Close button and then the Clear button to

remove the TCP stream filter.

\item Apply the display filter frame contains "X-Slogan"

\item Right-click on the two other displayed frames and select Follow | TCP Stream to

examine the HTTP headers exchanged between hosts. Did you find the other message?

Note that you can only follow one stream at a time using this right-click method. You

will need to clear out your display filter before following the next stream.

\item[5.] [10 pts] What other message did you find (different than Q4)?

\item Part 3 clean-up: Click the Close button on the Follow TCP Stream window when you

have finished following streams. Click the Clear button to remove your display filter

before continuing.

\end{itemize}

\subsection{Extract binary file from FTP session (15 pts)}

In the previous section, we extracted ASCII-text messages from packets. What about binary

data? Wireshark has tools for this as well.

\begin{itemize}

\item Open pcaps/ftp-clientside101.pcapng in Wireshark.

\item Scroll through the beginning of the trace file. You will see numerous FTP commands

used to login, request a directory, define a port number for the data transfer and

retrieve a file.

\item There are two data connection in this trace file: one for the directory list and another

for the file transfer. We are only interested in these two data streams, and not the

command channel stream. In the Follow TCP Stream window, click the Filter Out

This Stream button. This closes the TCP stream window and applies an exclusion

filter.

\item[6.] [5 pts] Now you only see the data channel traffic. Frames 16 through 18 and frames 35

through 38 are TCP handshake packets to establish the two required data channels.

Right-click on frame 16 and select Follow | TCP Stream. This stream list indicates

there is only one file in the directory. What is its name? (You will use it next.)

\item Click the Filter Out This Stream button. This closes the TCP stream window and

adds to the existing exclusion filter.

\item The only remaining traffic displayed is the file transfer traffic. Right-click on any frame

and select Follow | TCP Stream. You can view the file identifier that indicates this

is a .jpg file (JFIF) and the metadata contained in the graphic file.

\item To reassemble the graphic image transferred in this FTP communication, Change the

Show and save data as dropdown to “Raw” and click the Save As button, select

a target directory for the file, and set the file name to the one you found a few steps

back. Click Save.

\item[7.] [10 pts] Navigate to the target directory and open the file you saved in the previous

step. Include the image in your report.

\item Part 4 clean-up: When you’ve finished examining the image you extracted, close your

image viewer. Return to Wireshark to close the TCP Stream window and clear your

display filter

\end{itemize}

\section{Submission}

After finishing the lab, go to the terminal on your Linux system that was used to start the lab and type:

\begin{verbatim}

stoplab

\end{verbatim}

When you stop the lab, the system will display a path to the zipped lab results on your Linux system. Provide that file to

your instructor, e.g., via the Sakai site.

\section{References}

\begin{enumerate}

\item[[1]]Wireshark 101: Essential Skills for Network Analysis,

by Laura Chappell and Gerald Combs.

Published by Protocol Analysis Institute, 2013.

ISBN: 1893939723, 9781893939721

\end{enumerate}

\end{document}