LAB EXERCISE

3

COMP3134

Introduction to Cyber Security

Week: 4

Objective(s):
Demonstrate passive sniffing using WireShark

Learning Outcome(s):
Identify & baseline network traffic using network monitoring tools

Table of Contents

Contents

sSummary....3

A. Clone GitHub Repo....3

B. Server Firewall....3

C. Getting Started with WireShark....5

D. Running WireShark for First Time....5

E. Using WireShark....7

F. Install and Use tcpdump....8

G. Commit and Upload Changes to GitHub repo....8

Summary

Goal: Demonstrate passive sniffing using WireShark

In Effort To: Identify & baseline network traffic using network monitoring tools

A. Clone GitHub Repo

Clone course GitHub repo to any location on your local machine
Navigate to the location above and create a folder named **wk4**
Use this local folder created above to create all the files necessary for this Lab Exercise

B. Server Firewall

By default, there are many ports that are open of a typical Linux distribution default installation.
A firewall is a system that provides network security by filtering incoming and outgoing network traffic based on a set of user-defined rules.
The purpose of a firewall is to reduce or eliminate the occurrence of unwanted network communications while allowing all legitimate communication to flow freely.

Iptables

Iptables is a firewall program for Linux that will monitor traffic from and to your server using tables.
These tables contain sets of rules, called chains, that will filter incoming and outgoing data packets.
When a packet matches a rule, it is given a **target**, which can be another chain or one of these special values:

ACCEPT	Will allow the packet to pass through.
DROP	Will not let the packet pass through.
RETURN	Stops the packet from traversing through a chain and tell it to go back to the previous chain.

The three common **chains** are

INPUT	Controls incoming packets to the server.
FORWARD	Filters incoming packets that will be forwarded somewhere else.
OUTPUT	Filter packets that are going out from your server.

Installation of iptables

Most Linux distributions should have iptables installed by default.
To check to see if it is installed, run the following command:

```
man iptables
```

If it is not installed, run the following commands

```
apt-get update
apt-get install iptables
```

View all defined rules of iptables

```
iptables -L -v
```

Create a text files named **iptables_rules_1.txt** and copy all of the output of the command above

Manual for iptables

A manual of a command lists and describes all its options
The link to the manual page for iptables is the following:
https://linux.die.net/man/8/iptables

Using iptables

A valid iptables command takes the following syntax:

```
iptables -A <chain> -i <interface> -p <protocol (tcp/udp) >
-s <source> --dport <port no.>  -j <target>
```

The following are sample commands. Please use them as guides

Allow all connections on https protocol

iptables -A INPUT -p tcp --dport 443 -j ACCEPT

Disallow all traffic from IP address of 198.456.123

iptables -A INPUT -s 198.456.123 -j DROP

Remove all firewall rules and start with a clean slate

iptables -F

See all firewall runs and their line numbers
iptables -L --line-numbers

Remove a specified rule of the chain INPUT (*N* = integer line number)
iptables -D INPUT *N*

C. Getting Started with WireShark

What is WireShark?
Wireshark is a free and open-source packet analyzer. It is used for network troubleshooting, analysis, software and communications protocol development, and education.

How does Wireshark work?
Wireshark is a packet sniffer and analysis tool that captures network traffic on the local network and stores that data for offline analysis.
Wireshark captures network traffic from Ethernet, Bluetooth, Wireless (IEEE.802.11), Token Ring, Frame Relay connections, and more.

Note: A packet is a single message from any network protocol (i.e., TCP, DNS, etc.)

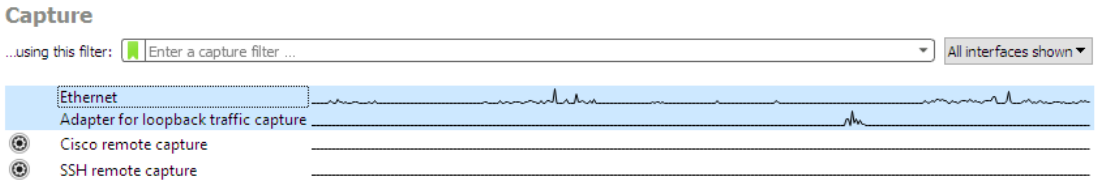
Installation
If you are using WireShark on your personal machines, please following the installation instructions below. WireShark is already installed on the lab machines

Please navigate to the link below to download and install WireShark

https://www.wireshark.org/download.html

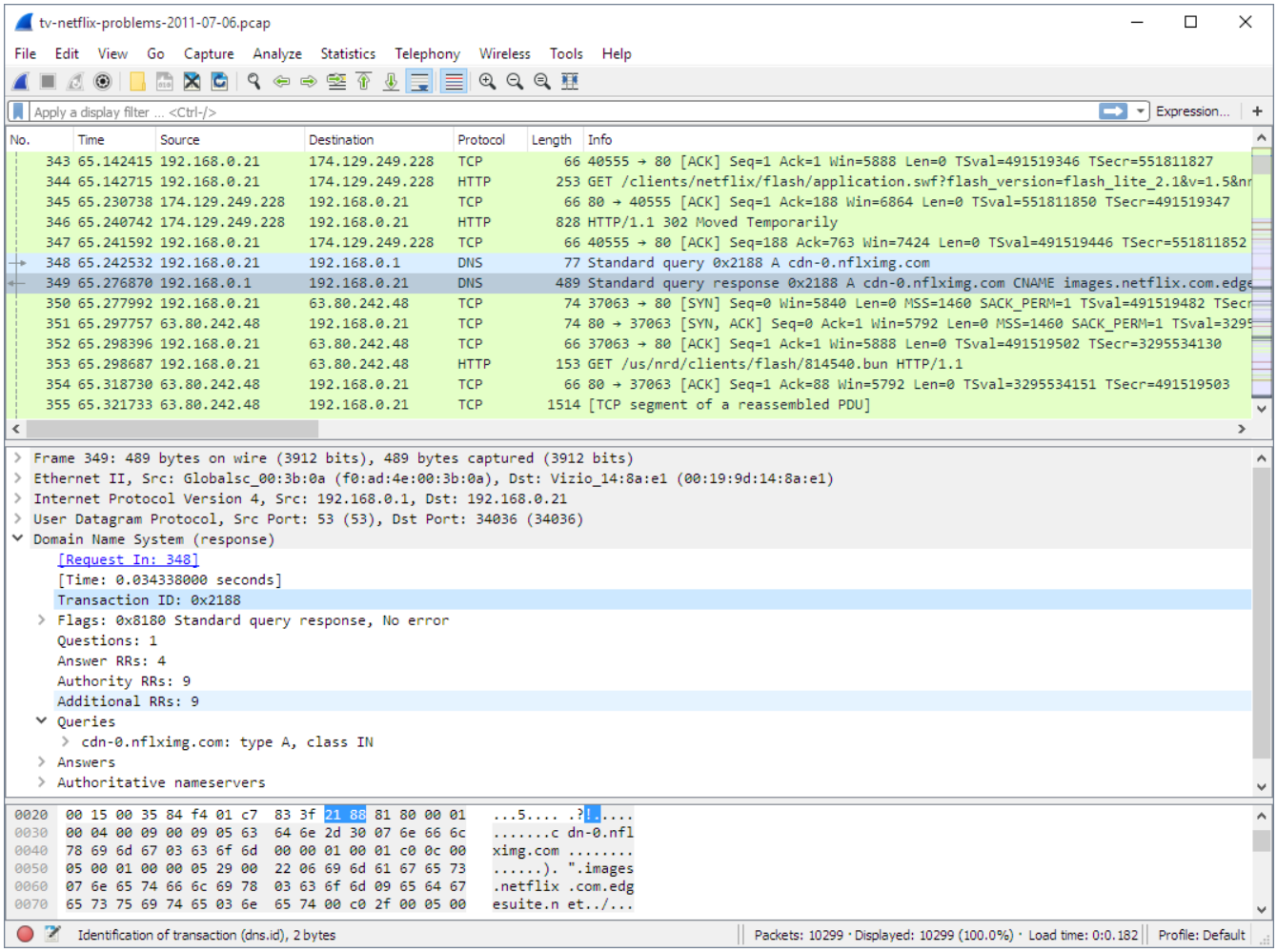
D. Running WireShark for First Time

Launching WireShark for first time
When you launch WireShark for the first time, you will see a welcome screen.
If you do not see this or you want to start a new capture, click on
Capture => Options from the menu



Select either Ethernet or Wi-Fi (depending on your internet connection) by double clicking on the option

You should now see the following screen



Create a screenshot named **wireshark_1.png** of your WireShark capture window

There are 7 components of the screen above

1. Menu
    a. https://www.wireshark.org/docs/wsug_html_chunked/ChUseMenuSection.html
2. Main Toolbar
    a. https://www.wireshark.org/docs/wsug_html_chunked/ChUseMainToolbarSection.html
3. Filter Toolbar
    a. https://www.wireshark.org/docs/wsug_html_chunked/ChUseFilterToolbarSection.html
4. Packet List Pane
    a. https://www.wireshark.org/docs/wsug_html_chunked/ChUsePacketListPaneSection.html
5. Packet Details Pane
    a. https://www.wireshark.org/docs/wsug_html_chunked/ChUsePacketDetailsPaneSection.html
6. Packet Bytes Pane
    a. https://www.wireshark.org/docs/wsug_html_chunked/ChUsePacketBytesPaneSection.html
7. Statusbar
    a. https://www.wireshark.org/docs/wsug_html_chunked/ChUseStatusbarSection.html

Display Filter

There may be times where you would like to capture all traffic but filter the captured traffic.
To accomplish this, navigate to the **Filter Toolbar**
Use the guide on the following page to understand how to best use the Filter Toolbar
https://wiki.wireshark.org/DisplayFilters#Examples

Capture Filters

There may be times where you'd like to only capture specific traffic.
To accomplish this, start a new capture by clicking on Capture => Options from the Menu
And follow the guide on the following page
https://wiki.wireshark.org/CaptureFilters#Capture_filter_is_not_a_display_filter

Switch Captures

Stop the current Capture by clicking on Capture => Stop
Start a new Capture by clicking on Capture => Start

Select another Capture Input Interface. (you may see no traffic)
Start the new Capture

Create a screenshot named **wireshark_2.png** of your new WireShark capture window

E. Using WireShark

Execute the following steps using WireShark. For each step, take a screenshot name **using_wireshark_N.png**, where N represents the step number.

1) Filter to only show ARP or DNS packet traffic (no other traffic)
2) Filter to only show UDP or TCP packet traffic (no other traffic)
3) Filter to show TCP packet traffic on any port greater than 3000
4) Filter to show only HTTP packet traffic
5) Filter to how only HTTP post method request packet traffic
   (use a web page form so you can see at least one entry
6) Filter to show UDP packet traffic that contain the strong "google"
7) Filter to show HTTP packet traffic where the page was not found
8) Filter to show TCP packet traffic where the acknowledgement flag was set
9) Select any two IP addresses and show the Flow Graph demonstrating a 3-way handshake between these two IP addresses

F. Install and Use tcpdump

Tcpdump is a command line packet sniffer

Installation

Tcpdump may already come installed in your Linux distribution. In case not, to install tcpdump, type the following command

```
apt-get install tcpdump
```

Using tcpdump

The following link provides a guide on how to use tcpdump.
Create text files that show the command and result of each step with the filename format of **tcmpdump_step_*N*.text** where N represents the step number
Start with #1.
https://www.tecmint.com/12-tcpdump-commands-a-network-sniffer-tool/

G. Commit and Upload Changes to GitHub repo

Commit the changes to your repo by:

1. Opening a GitBash window and ensure that it is connected to your local machine
2. Navigate to local repository directory location
3. Add all the files completed in this Lab Exercise
4. Commit the changes
5. Push the changes to your GitHub course repo

Please refer to the instructions in the last section of Lab Exercise 1