

Name:	Student #:	Group:
--------------	-------------------	---------------

CSP Lab #2: Secure Network Communication

Required Resources:

- Kali Linux
- Internet connection

Learning Objectives:

Part 1: Using Telnet and SSH in the Linux terminal

Part 2: Capturing and analysing traffic using Wireshark. (50%)

Part 3: Using Tcpdump, Linux tools options, and man pages (30%)

Part 4: Secure web browsing, more Wireshark filters, driftnet.

Instructions

You are required to submit a written report of how you accomplished the tasks in this laboratory. The report should consist of a series of screen shots and a written account of your observations. The report must be submitted to Blackboard within a week of completing the laboratory session. Standard CIT penalties for late submission will apply.

Part 1: Using Telnet, SSH, and FTP

Telnet and Secure Shell (SSH) are protocols used for remote administration. Both protocols allow a user to log into a system with a username and passwords and run commands on that system or access a service hosted there. File Transfer Protocol (FTP) allows a user to upload/download (pull/push) files to and from a server.

First, power on your Kali Linux machine/VM and log in (default credentials are “user: root, pass: toor”). Then open a terminal. The terminal application can usually be found under “Favorites” in the applications menu.

Telnet

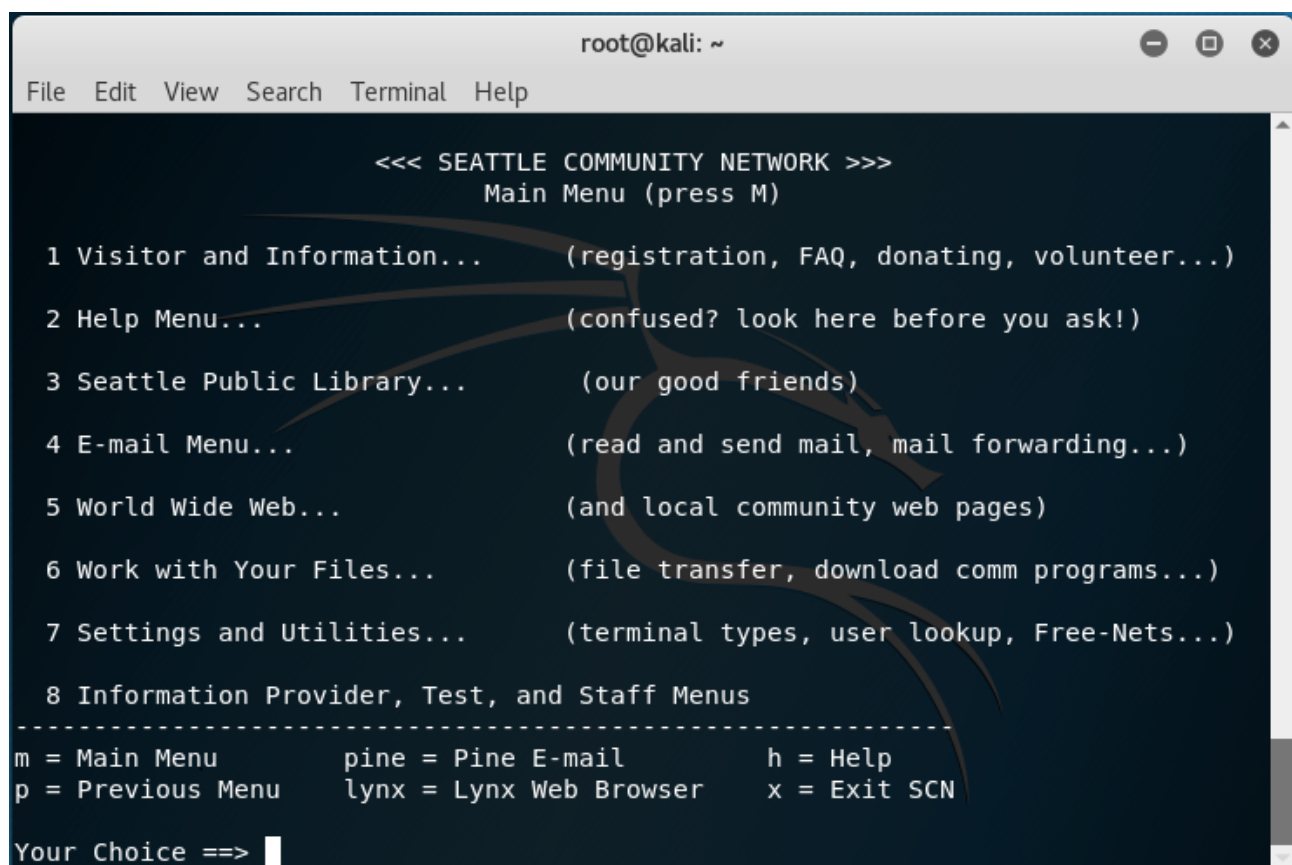
A Telnet client is available by default on Kali Linux. The server you’ll be connecting to is designed to be a community hub for Seattle, USA. This server doesn’t let you run commands, but instead offers a menu driven application that supplies information for the local community. The following command launches the Telnet application and connects to port 23 on the server “scn.org”.

In the terminal type:

```
telnet scn.org
```

Use the username “visitor” to log in.

Follow the directions on-screen to reach the main menu and from there use “x” to exit.



```
root@kali: ~
File Edit View Search Terminal Help

<<< SEATTLE COMMUNITY NETWORK >>>
Main Menu (press M)

1 Visitor and Information... (registration, FAQ, donating, volunteer...)
2 Help Menu... (confused? look here before you ask!)
3 Seattle Public Library... (our good friends)
4 E-mail Menu... (read and send mail, mail forwarding...)
5 World Wide Web... (and local community web pages)
6 Work with Your Files... (file transfer, download comm programs...)
7 Settings and Utilities... (terminal types, user lookup, Free-Nets...)
8 Information Provider, Test, and Staff Menus
-----
m = Main Menu      pine = Pine E-mail      h = Help
p = Previous Menu  lynx = Lynx Web Browser  x = Exit SCN

Your Choice ==> 
```

SSH

Like Telnet, Kali has an SSH client available by default. For the SSH connection we’ll use a public SSH server. The following command will launch the SSH client, and connect to port 22 on the

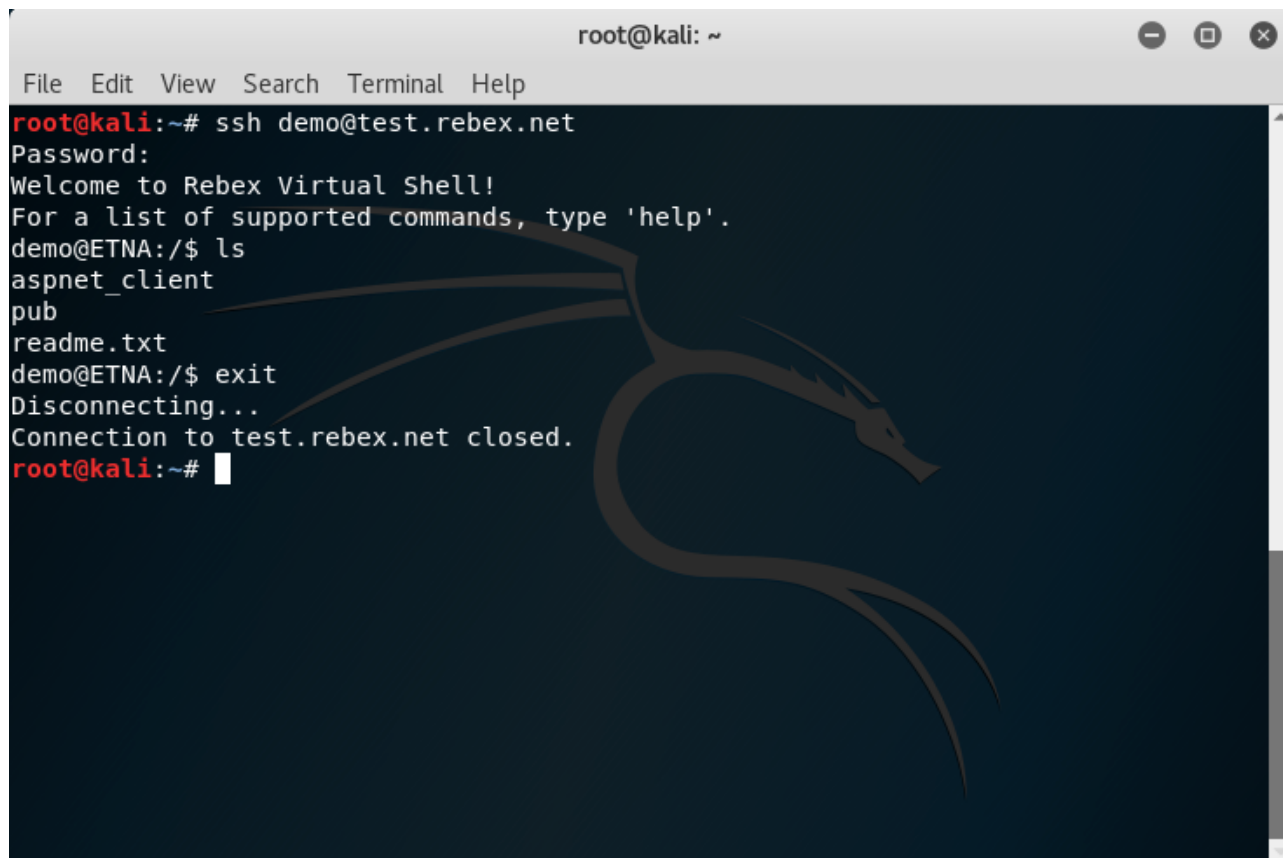
“test.rebex.net” server. The “demo@” pre-appended to the server name specifies the user we wish to log in as.

Note: If you are prompted to remember the remote host simply type “yes”.

```
ssh demo@test.rebex.net
```

Once connected you will be asked for a password. Use the password “password” to log in.

The server offers a limited number of commands which may be used. Type “ls” to list the directory contents. Type “exit” to disconnect.

A screenshot of a terminal window titled "root@kali: ~". The terminal shows the command "ssh demo@test.rebex.net" being executed. It prompts for a password, then displays a welcome message from the "Rebex Virtual Shell". The user runs "ls" and sees a list of files: "aspnet_client", "pub", and "readme.txt". Finally, the user runs "exit", and the terminal shows "Disconnecting..." and "Connection to test.rebex.net closed." before returning to the root@kali prompt. A large, faint dragon logo is visible in the background of the terminal window.

```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# ssh demo@test.rebex.net  
Password:  
Welcome to Rebex Virtual Shell!  
For a list of supported commands, type 'help'.  
demo@ETNA:/$ ls  
aspnet_client  
pub  
readme.txt  
demo@ETNA:/$ exit  
Disconnecting...  
Connection to test.rebex.net closed.  
root@kali:~#
```

Services such as the one seen with the Telnet connection can be offered over SSH (See the final section for links to public Telnet & SSH services). One important difference between Telnet and SSH is that traffic sent over Telnet is unencrypted while traffic sent over SSH is encrypted. Furthermore, SSH clients and servers can be validated using certificates, making it very difficult for an attacker to impersonate a particular server.

Part 2: Capturing and analysing traffic using Wireshark (50%)

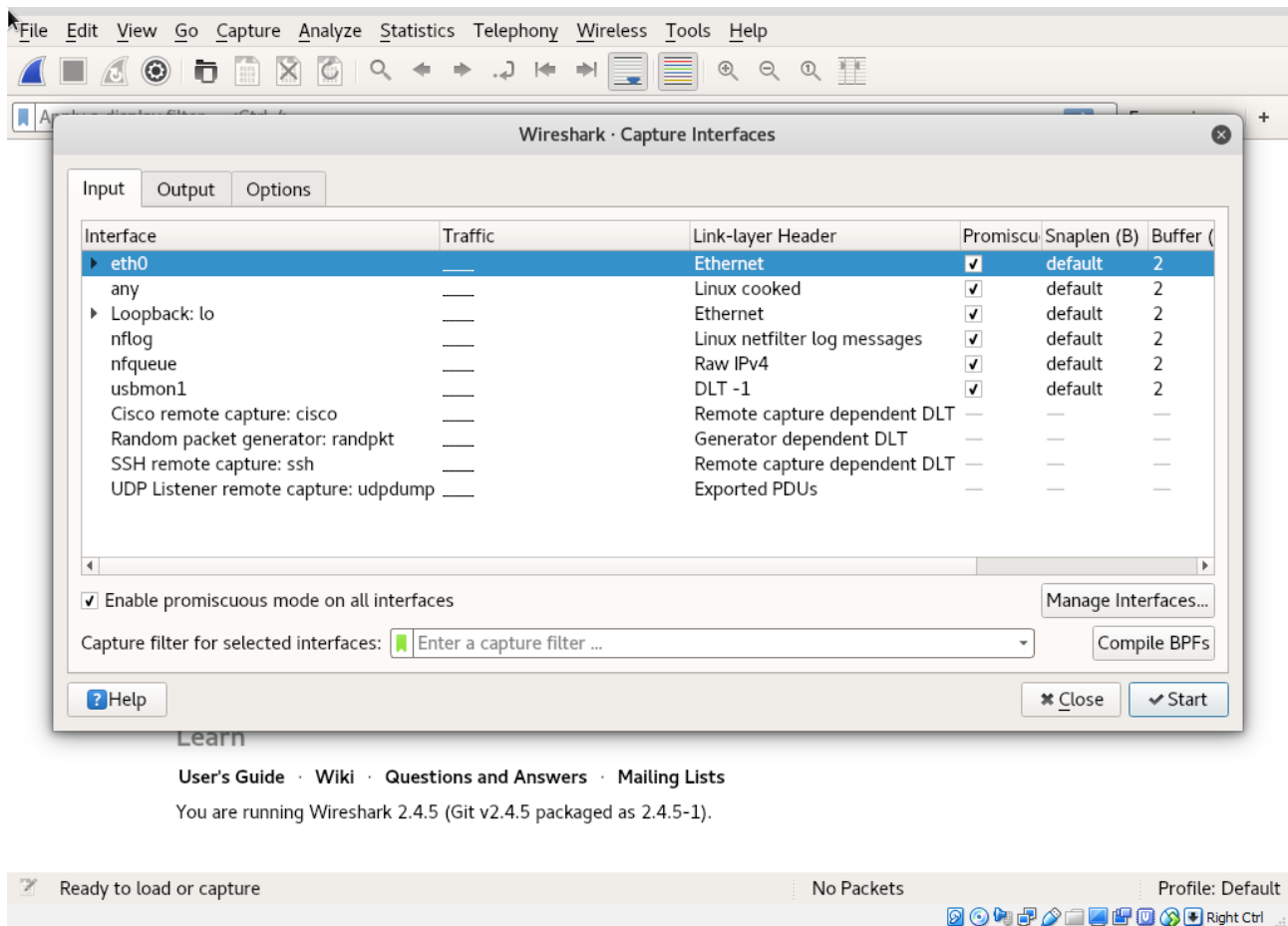
Wireshark is a tool used to capture and analyse network traffic. Using this tool, you will be able to see the difference between Telnet and SSH. This tool can be found under the “Sniffing & Spoofing” section of Kali’s applications menu.

Selecting an Interface

Before capturing traffic, you will have to specify the network interface you wish Wireshark to

capture traffic for.

To select an interface and start the traffic capture, navigate to “capture → options” on the main toolbar.



To simplify things for this lab you will capture traffic on all interfaces. To do this, just select “any” as the interface and click “Start” to continue. Wireshark is now capturing traffic.

Capturing Telnet and SSH Traffic

With Wireshark capturing traffic, repeat the steps in part 1. Once the steps have been completed, navigate to “capture → stop” in the main toolbar. Clicking on stop will stop the current packet capture. The packets Wireshark has captured will contain packets from both the Telnet and SSH connection.

Viewing the Captured Traffic

The Wireshark window is divided into 3 rows, or panels. The topmost panel contains one-line synopsis of captured packets. The type of information seen here is source, destination, protocol, length, and a short snippet of information. The contents of this panel can be scrolled through and clicked on. Once a packet in this panel has been clicked and selected, the middle panel will display detailed information on each protocol included in the packet. The bottom panel will display the packet in hexadecimal and show the ASCII representation where possible.

Wireshark capture showing a Telnet session. The packet list shows a SYN packet (No. 30) and several data packets. The packet details pane shows the selected packet (No. 30) with fields: Ethernet II, Internet Protocol Version 4, Transmission Control Protocol, and VSS-Monitoring ethernet trailer. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
30	623.724614602	66.212.64.194	10.0.2.15	TCP	62	23 → 54092 [SYN, ACK] Seq=0
31	623.724636121	10.0.2.15	66.212.64.194	TCP	56	54092 → 23 [ACK] Seq=1 Ack=1
32	623.725038802	10.0.2.15	66.212.64.194	TELNET	83	Telnet Data ...
33	623.727233502	66.212.64.194	10.0.2.15	TCP	62	23 → 54092 [ACK] Seq=1 Ack=2
34	624.134545461	66.212.64.194	10.0.2.15	TELNET	62	Telnet Data ...
35	624.134557336	10.0.2.15	66.212.64.194	TCP	56	54092 → 23 [ACK] Seq=28 Ack=
36	624.341044421	66.212.64.194	10.0.2.15	TELNET	184	Telnet Data ...
37	624.341058598	10.0.2.15	66.212.64.194	TCP	56	54092 → 23 [ACK] Seq=28 Ack=
38	624.341170544	10.0.2.15	66.212.64.194	TELNET	76	Telnet Data ...

Frame 30: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface 0
 Linux cooked capture
 Internet Protocol Version 4, Src: 66.212.64.194, Dst: 10.0.2.15
 Transmission Control Protocol, Src Port: 23, Dst Port: 54092, Seq: 0, Ack: 1, Len: 0
 VSS-Monitoring ethernet trailer, Source Port: 0

```

0000  00 00 00 01 00 06 52 54 00 12 35 02 00 00 08 00  .....RT ..5....
0010  45 00 00 2c 02 ad 00 00 40 06 e8 7a 42 d4 40 c2  E.,... @..zB.@.
0020  0a 00 02 0f 00 17 d3 4c 00 1c 52 01 c0 ee 51 4d  .....L ..R...QM
0030  60 12 ff ff d0 b4 00 00 02 04 05 b4 00 00  .....
  
```

wireshark_any_20181030182736_uu4Lxl Packets: 46 · Displayed: 46 (100.0%) · Dropped: 0 (0.0%) Profile: Default

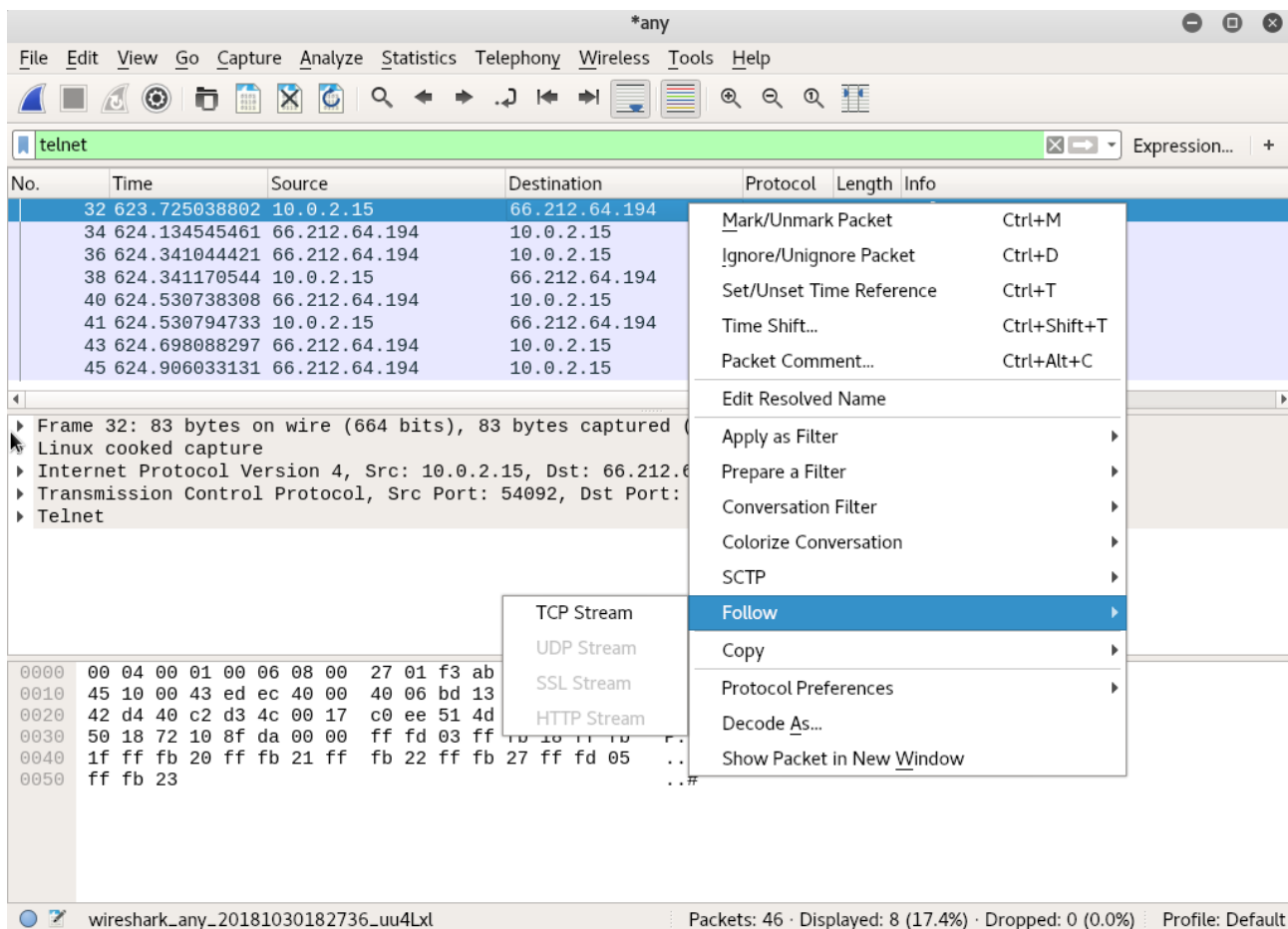
Filtering and Viewing Data

Wireshark allows filters to be used to help in analysing traffic. Filters can be typed into the bar just below the main toolbar. Typing in “telnet” will show traffic relevant to Telnet, and typing “ssh” will filter to only SSH traffic.

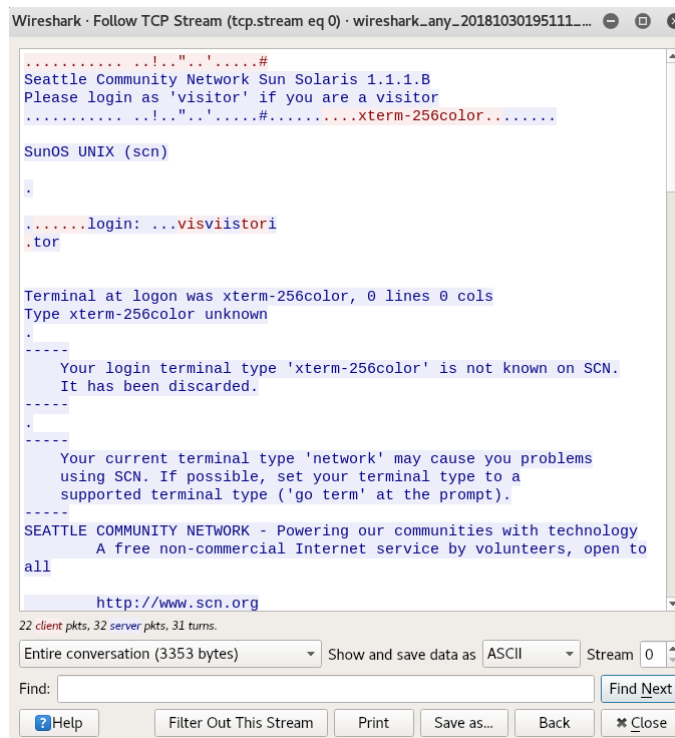
Wireshark capture showing a filtered view of Telnet traffic. The filter bar at the top contains the text “telnet”. The packet list shows only Telnet data packets.

No.	Time	Source	Destination	Protocol	Length	Info
32	623.725038802	10.0.2.15	66.212.64.194	TELNET	83	Telnet Data ...
34	624.134545461	66.212.64.194	10.0.2.15	TELNET	62	Telnet Data ...
36	624.341044421	66.212.64.194	10.0.2.15	TELNET	184	Telnet Data ...
38	624.341170544	10.0.2.15	66.212.64.194	TELNET	76	Telnet Data ...
40	624.530738308	66.212.64.194	10.0.2.15	TELNET	90	Telnet Data ...
41	624.530794733	10.0.2.15	66.212.64.194	TELNET	62	Telnet Data ...
43	624.698088297	66.212.64.194	10.0.2.15	TELNET	63	Telnet Data ...
45	624.906033131	66.212.64.194	10.0.2.15	TELNET	62	Telnet Data ...

It’s also possible to filter to a specific conversation and view the data sent during that conversation. To do this, right click on one of the Telnet packets, navigate to “Follow”, and select “TCP Steam”.



Wireshark will display the data from the conversation in ASCII representation.



Perform the same action with the captured SSH traffic.

By clicking through the captured Telnet packets and observing the data displayed in the bottom panel, can you find the packet which contains the text from the scn.org application's main menu? Take a screenshot of this packet highlighted, and with the menu data showing in the bottom panel of Wireshark. Add this to your report.

Searching through the SSH traffic, can you find the packet which contained the directory content data? Add details of your answer to your report.

What is the main security issue you observe with the Telnet traffic? Add details (including screenshots) of this to your report.

Part 3: Using Tcpdump (30%)

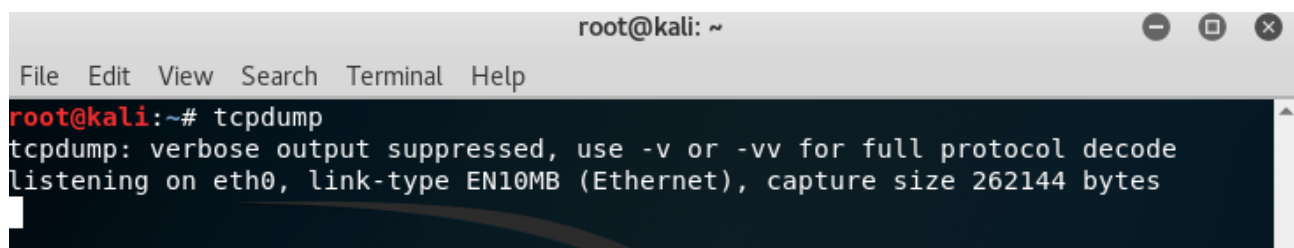
Another useful tool for capturing traffic is Tcpdump. This tool works in the terminal, and unlike Wireshark, doesn't require any desktop environment. This is useful in situations where you may have connected to a remote server using SSH and want to capture traffic.

Running Tcpdump

Open a terminal window and use the following command to launch Tcpdump:

```
tcpdump
```

Tcpdump will launch and capture on interface "eth0" by default.

A screenshot of a terminal window titled "root@kali: ~". The terminal shows the command "tcpdump" being entered. Below the command, a message is displayed: "tcpdump: verbose output suppressed, use -v or -vv for full protocol decode listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes". The terminal has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help".

```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# tcpdump  
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode  
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
```

Flags, or options, can be passed to Tcpdump to allow things such as the interface to be specified. The available options can be seen by using "--help" after the tcpdump command like so:

```
tcpdump --help
```

A manual page is also available for Tcpdump, which can be access using the following command:

```
man tcpdump
```

Type "q" to quit.

Using the information from the help text and the man page, it can be seen that the "-i" option is used to specify the interface Tcpdump should capture on. The following command will tell Tcpdump to capture on the localhost interface:

```
tcpdump -i lo
```


Adding "icmp" to the end of the command will have Tcpdump filter for ICMP (ping) traffic. Use Tcpdump to capture on the localhost interface, filter for pings, and print packets in ASCII representation (*Hint: this option can be found using the Tcpdump man page*). Add this command to your report and include an explanation of each option used.

In another terminal window, use the following command to send a ping:

```
ping -c 1 127.0.0.1
```

Using the output from Tcpdump, identify the payload (data) of the ping that you sent. Include a screenshot of this in your report.

Use Wireshark to capture the ping and take a screenshot of the ICMP payload.

Part 4: (Challenge) Secure Web Browsing (20%)

Using Wireshark, capture traffic between Kali and the following sites:

<http://www.1112.net/lastpage>.

<https://www.warnerbros.com/>

In part 2, the "telnet" filter was used to filter traffic in Wireshark. What filters could be used to distinguish between the connections to the two websites?

What difference can you observe between the two connections? Can you view the data from the web pages in Wireshark?

Kali has a tool called "driftnet" installed. Using <http://www.cit.ie> demonstrate how this tool can be used. Does this tool work for every website? If not, why not?

Document your answers to the above questions in your report. Use screenshots to support your answers.

Additional Resources:

Small collection of PCAP files, including packet captures from Malware:

<https://www.netresec.com/?page=PcapFiles>

More PCAP files:

<https://wiki.wireshark.org/SampleCaptures>

Watch Star Wars over Telnet (Yes, really)

towel.blinkenlights.nl

Other places to telnet

<https://www.telnet.org/htm/places.htm>

<https://www.jumpjet.info/Offbeat-Internet/Public/TelNet/url.htm>

