

## Abstract

This assignment aims to use Wireshark to intercept unencrypted packages sent over a network connection. After starting up Wireshark and sending packets using Netcat, they will be visible in the Wireshark user interface. This will demonstrate how any packet is vulnerable to being read by a human in case it is not encrypted.

## Introduction

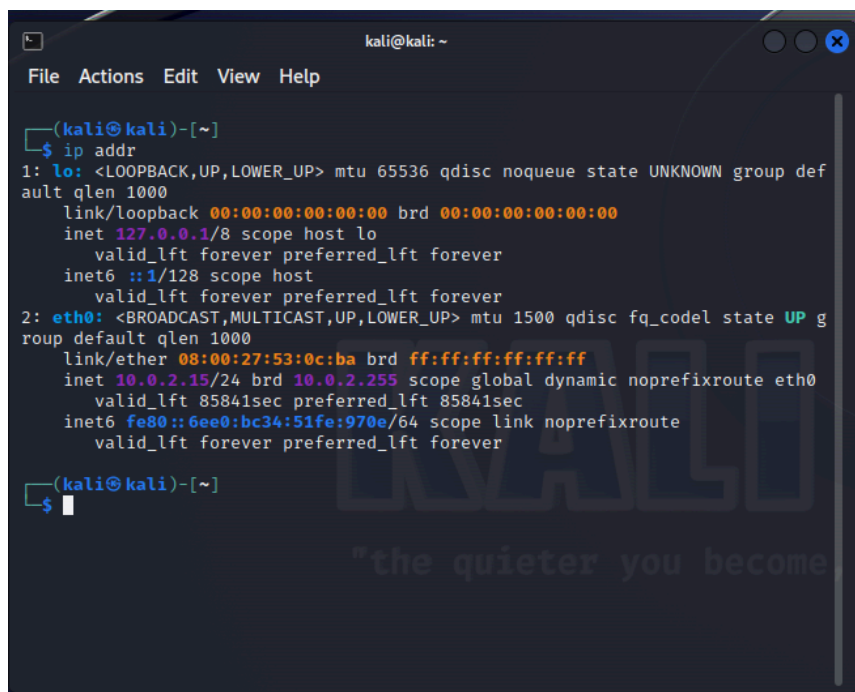
Using a VirtualBox machine, Kali Linux will be booted up. Netcat will be used to set up to send and receive data, and Wireshark will be used to inspect the packets. This will be done using the following Netcat commands:

'nc -l 10.0.2.15 -p 5656' to listen for packets,  
And 'nc 10.0.2.15 5656' to send packets.

Wireshark will listen to all packets on the 'any' adapter.

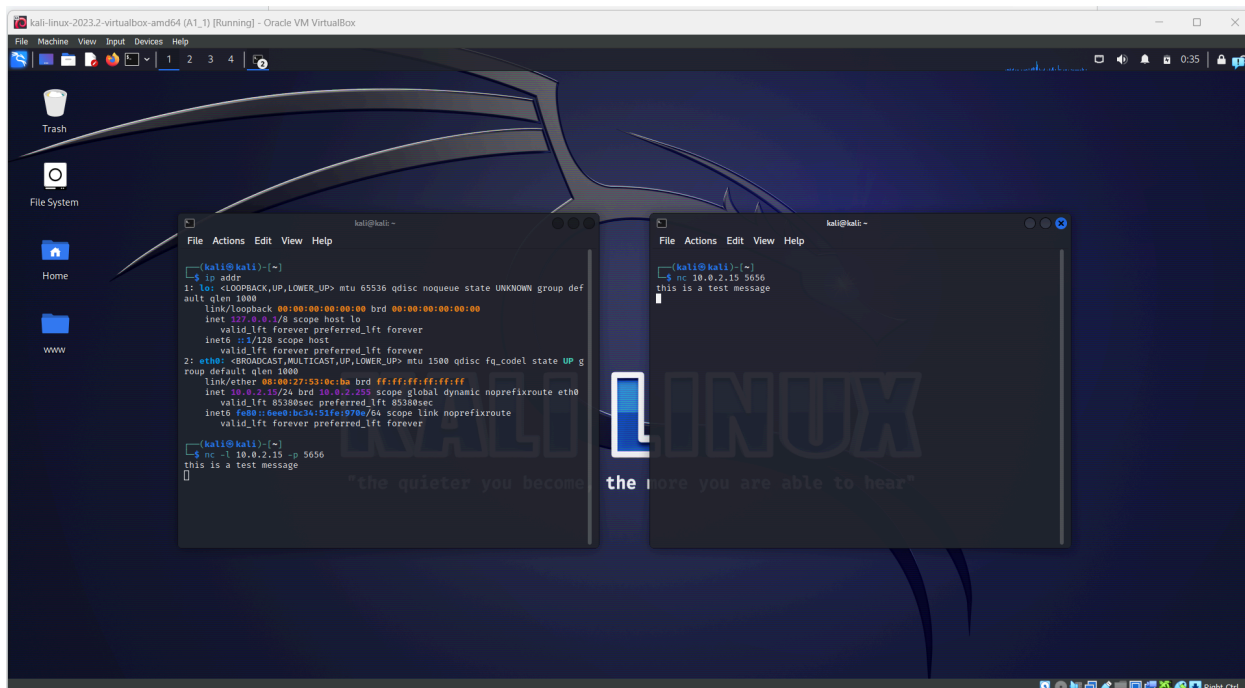
## Summary of Results

The first thing done is to open the Terminal to check for the machine's IP address. Since this is a preconfigured virtual machine, the local address is '10.0.2.15' which can be checked using the 'ip addr' command.

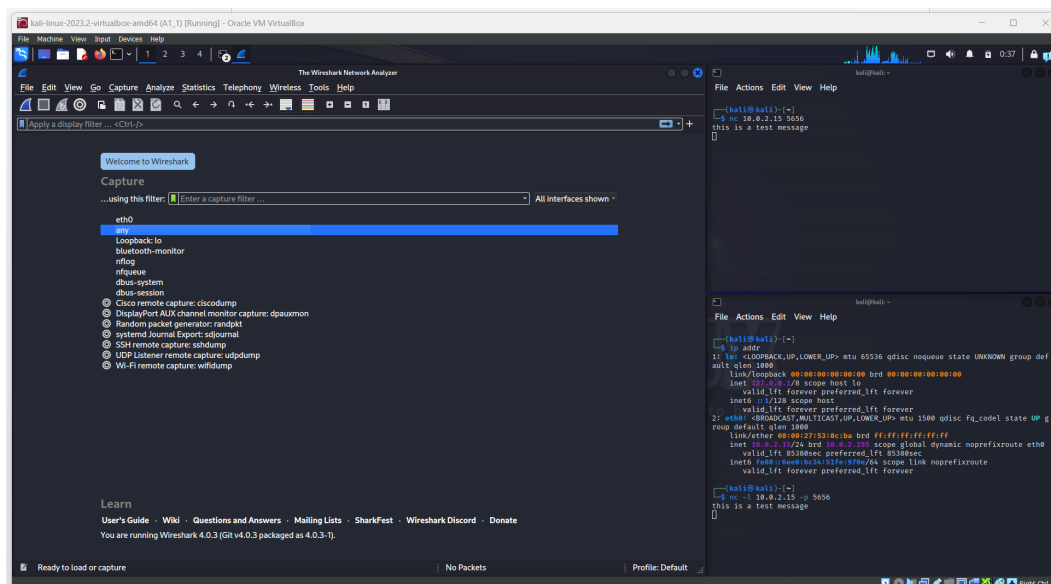


```
kali@kali: ~  
File Actions Edit View Help  
  
(kali@kali)-[~]  
$ ip addr  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group def  
ault qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host  
        valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g  
roup default qlen 1000  
    link/ether 08:00:27:53:0c:ba brd ff:ff:ff:ff:ff:ff  
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute eth0  
        valid_lft 85841sec preferred_lft 85841sec  
    inet6 fe80::6ee0:bc34:51fe:970e/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever  
  
(kali@kali)-[~]  
$
```

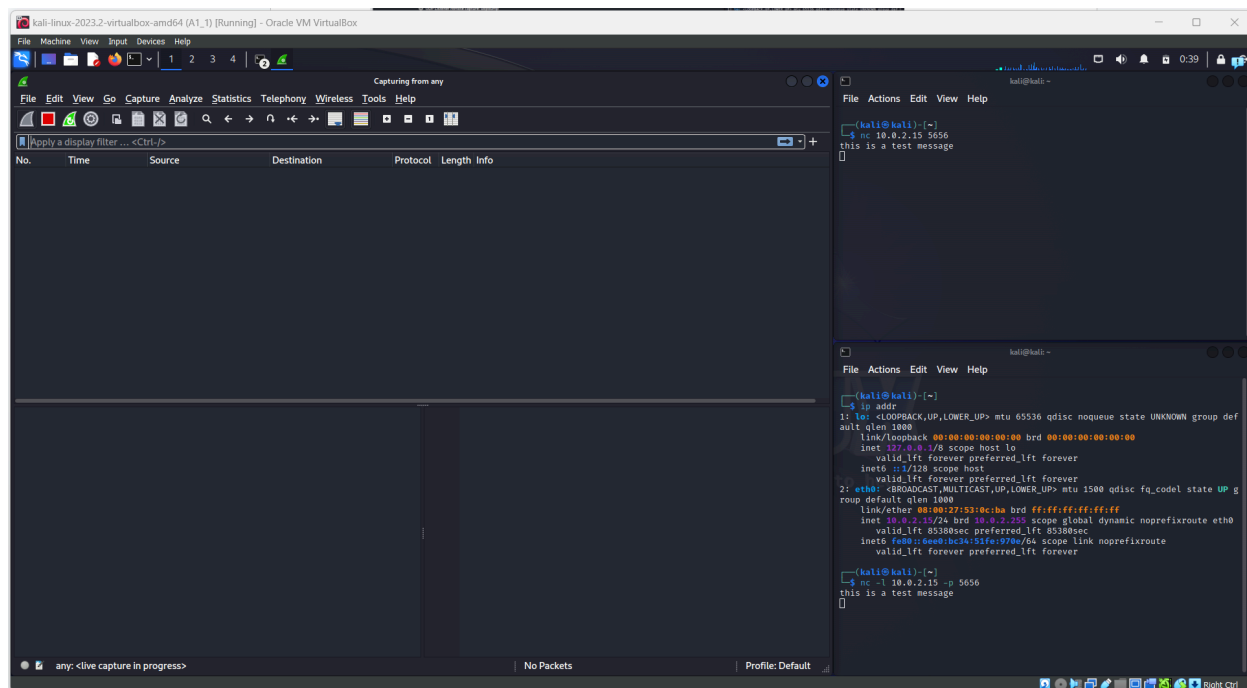
Then a second Terminal window is opened. Netcat will be used to set up communication between the two terminals over a local network. Messages sent over the two networks will be unencrypted and will be used as a packet example for Wireshark. One terminal window will be used to listen to packets, while the other is used to send them. After typing in the commands presented in the introduction, the two terminals connect. We can see that by typing in a message into the terminal that sends data and seeing it appear on the terminal that listens to data.



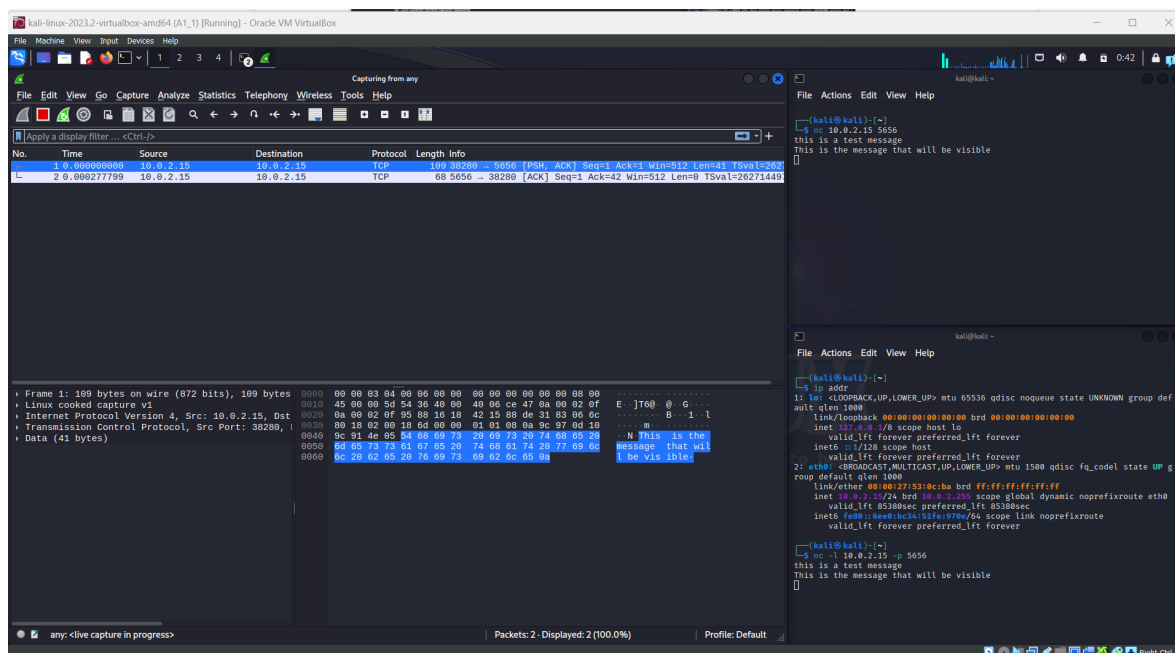
A Wireshark window is then opened, and the 'any' adapter is selected from the list by double-clicking on it.



After this step, Wireshark will start capturing packets on the network. As of this moment, there will be nothing visible as nothing was sent yet.



Another message is now sent through Netcat by typing it into the sender Terminal. This will produce two packets visible in the Wireshark interface. Upon selecting the first packet, its contents will be displayed in the bottom right portion of Wireshark in plain text as well as hexadecimal numbers. The last part of the message will contain the message that was sent through Netcat, which proves that unencrypted traffic is visible to anyone who might be listening to the network.



(Since I didn't correctly produce my output file the first time, I went back and repeated the process using the same method and message. The initial parts of the packets in the screenshot and the output file do not fully match up for that reason.)

## Conclusion

This process shows how easy it is to use a program such as Wireshark to listen to traffic on the Network. If an attacker were to use a program like Wireshark on a network that wasn't a private Virtual Machine network, they would be able to see any unencrypted data with proper package filtering, which puts users of the network at a privacy risk. This is the reason proper encryption is needed as a step of ensuring data is safe from attackers, as it makes the packets unreadable by a human, and would require other methods of accessing the contents of it.

- Describe Wireshark in your own words:

Wireshark is a free, open-source program used to listen to and analyze packets sent over a network. It displays all packets sent over the network in a GUI allowing users to easily select, filter, and export the history of the chosen packets.

- How is this a useful tool?

Wireshark can be a useful tool for a few reasons depending on what the user is trying to get out of it. For some, it can be a great tool to discover potential network and data risks, learn how one's network behaves, discover suspicious network communication, as well as troubleshoot network problems. However, if it was accessed by an attacker, it could be used to spy on network traffic and gather confidential information that was unencrypted.

- What were you able to discover using it?

Using Wireshark, I was able to discover the encrypted packets of data I generated. My messages were visible in plain text, and if they were confidential, this would have been a breach of confidentiality. I also discovered how easy it would have been for another person to intercept those same messages had they had access to the private network.