Jorel Gregg

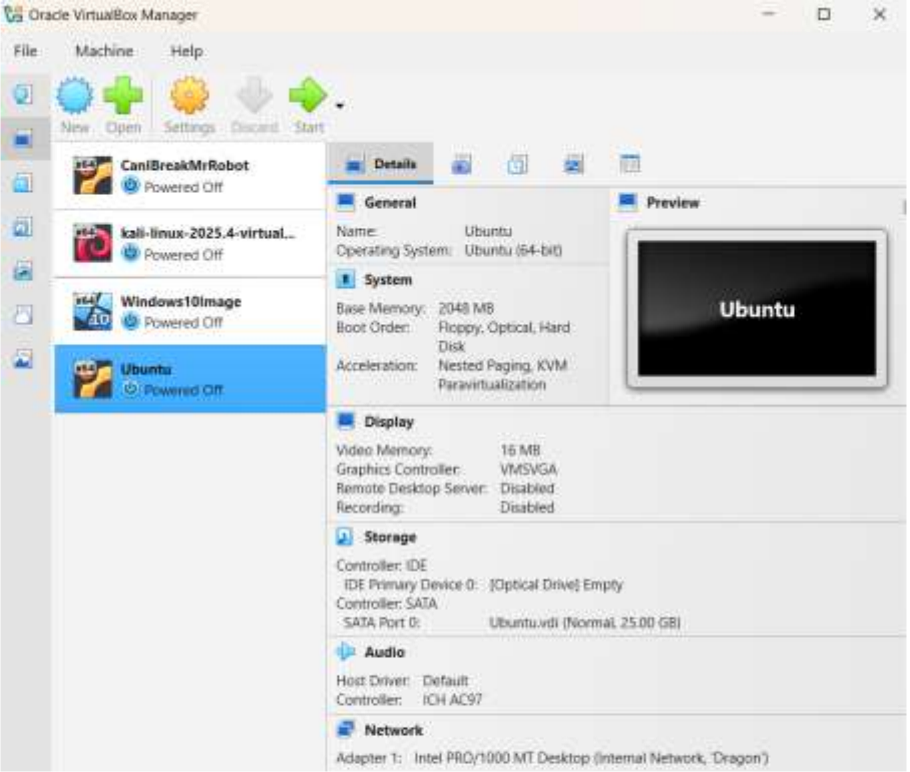
1/14/26

Initial Wireshark Home Lab

I built a virtual network lab using VirtualBox, Kali-Linux, Ubuntu, and then used Wireshark to analyze network traffic, identify insecure protocols, capture authentication exchanges, detect suspicious behavior, and document findings. I used Wireshark to compare encrypted vs unencrypted protocols and demonstrated how attackers can extract credentials from FTP traffic. I troubleshot packet capture issues related to host-only networking, Windows permissions, and Npcap drivers. I configured dual-homed virtual machines to securely enable package installation while keeping lab traffic isolated. I managed Linux user accounts and reset credentials while troubleshooting service authentication. I used vsftpd to host an FTP service in a virtual lab and demonstrated how cleartext credentials can be captured using Wireshark. I observed TCP SYN scanning behavior from Kali Linux targeting Ubuntu Server, characterized by repeated SYN packets to sequential ports without full TCP handshakes, consistent with Nmap reconnaissance activity.

Steps:

1. I installed Ubuntu-Server on a new VM in VirtualBox and set the network to the host created network by VirtualBox.



CanIBreakMrRobot
Powered Off

kali-linux-2025.4-virtual...
Powered Off

Windows10Image
Powered Off

Ubuntu
Powered Off

General

Name: Ubuntu
Operating System: Ubuntu (64-bit)

System

Base Memory: 2048 MB
Boot Order: Floppy, Optical, Hard Disk
Acceleration: Nested Paging, KVM Paravirtualization

Display

Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage

Controller: IDE
IDE Primary Device 0: [Optical Drive] Empty
Controller: SATA
SATA Port 0: Ubuntu.vdi (Normal, 25.00 GB)

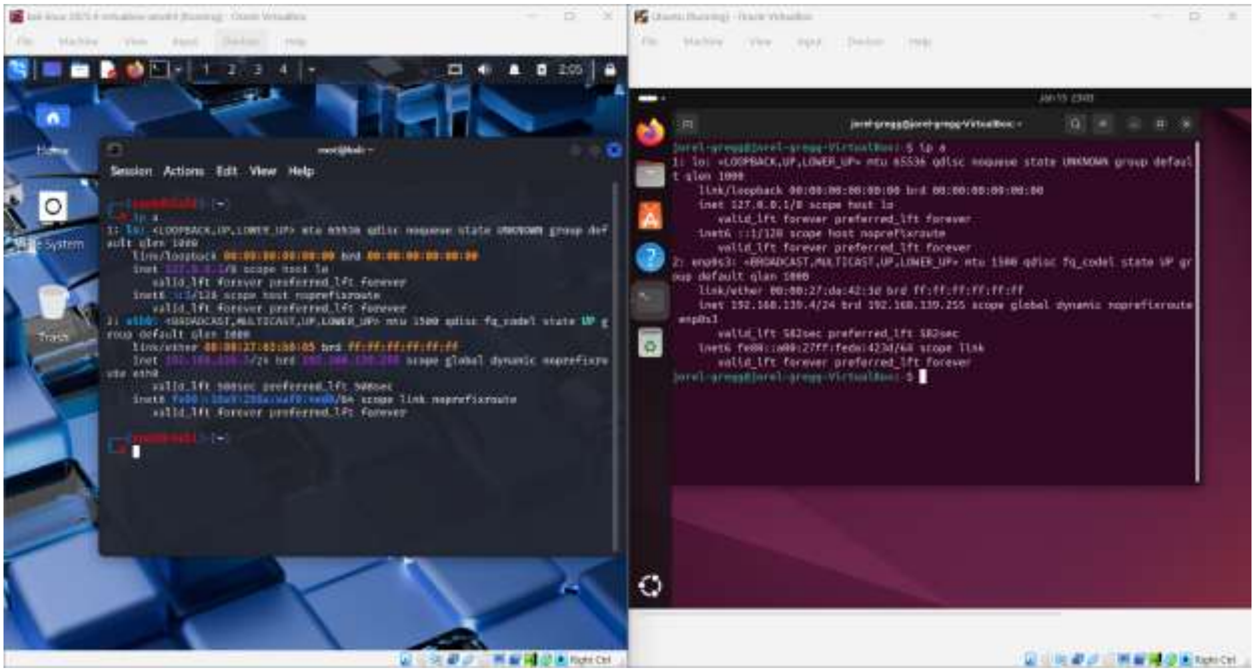
Audio

Host Driver: Default
Controller: ICH AC97

Network

Adapter 1: Intel PRO/1000 MT Desktop (Internal Network, 'Dragon')

2. Next, I looked up the IP addresses that the host network had assigned my two machines to enable an easier time of just capturing packets via Wireshark.



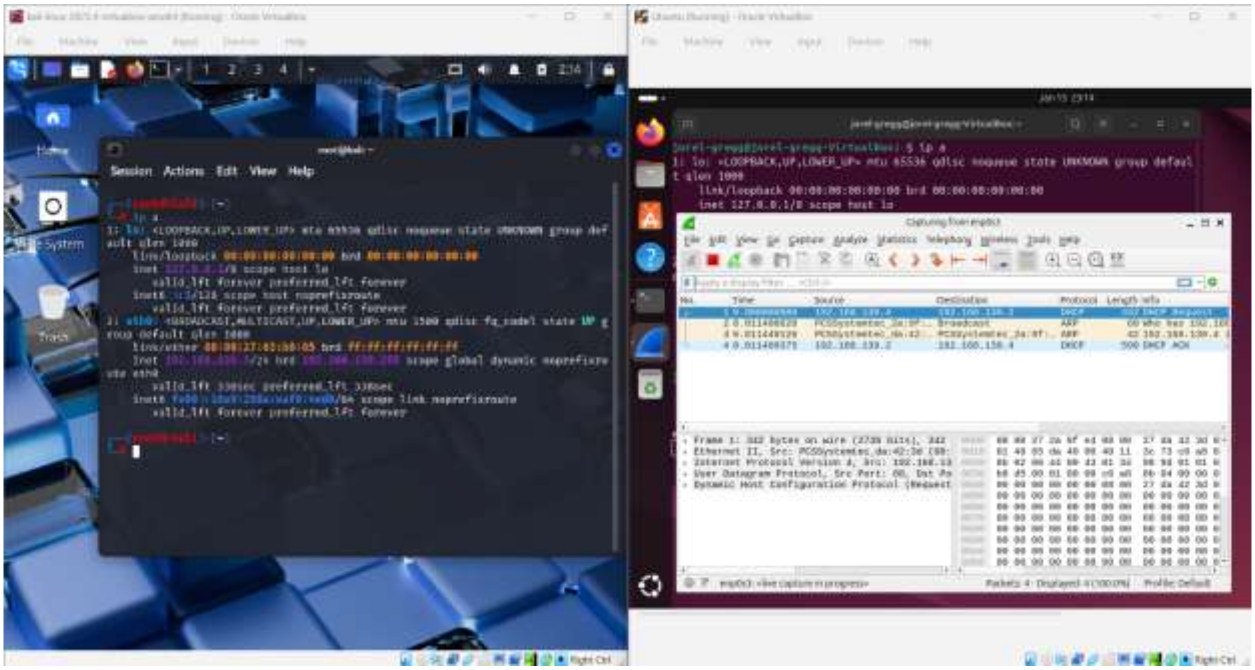
```
root@kali: ~
Session  Actions  Edit  View  Help

┌──([illegible])-[~]
└─# ip a
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 noprefixroute
       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 08:00:27:63:b0:05 brd ff:ff:ff:ff:ff:ff
    inet [illegible]/24 brd [illegible] scope global dynamic noprefixro
ute eth0
       valid_lft 508sec preferred_lft 508sec
    inet6 fe80::[illegible]/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

┌──([illegible])-[~]
└─#
```

```
jorel-gregg@jorel-gregg-VirtualBox:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t 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 noprefixroute
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 08:00:27:da:42:3d brd ff:ff:ff:ff:ff:ff
    inet 192.168.139.4/24 brd 192.168.139.255 scope global dynamic noprefixroute
 enp0s3
       valid_lft 582sec preferred_lft 582sec
    inet6 fe80::a00:27ff:feda:423d/64 scope link
       valid_lft forever preferred_lft forever
jorel-gregg@jorel-gregg-VirtualBox:~$
```

3. Then I ran Wireshark on the Ubuntu server and selected the host network for my machines to begin capturing traffic. This consisted of basic ARP and DHCP traffic packets which are just asking who has this IP address and then assigning that IP address.



```
root@kali: ~
Session  Actions  Edit  View  Help

┌──([illegible])-[~]
└─# ip a
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 noprefixroute
       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 08:00:27:63:b0:05 brd ff:ff:ff:ff:ff:ff
    inet [illegible]/24 brd [illegible] scope global dynamic noprefixro
ute eth0
       valid_lft 338sec preferred_lft 338sec
    inet6 fe80::[illegible]/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

┌──([illegible])-[~]
└─#
```

```
jorel-gregg@jorel-gregg-VirtualBox:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t 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
```

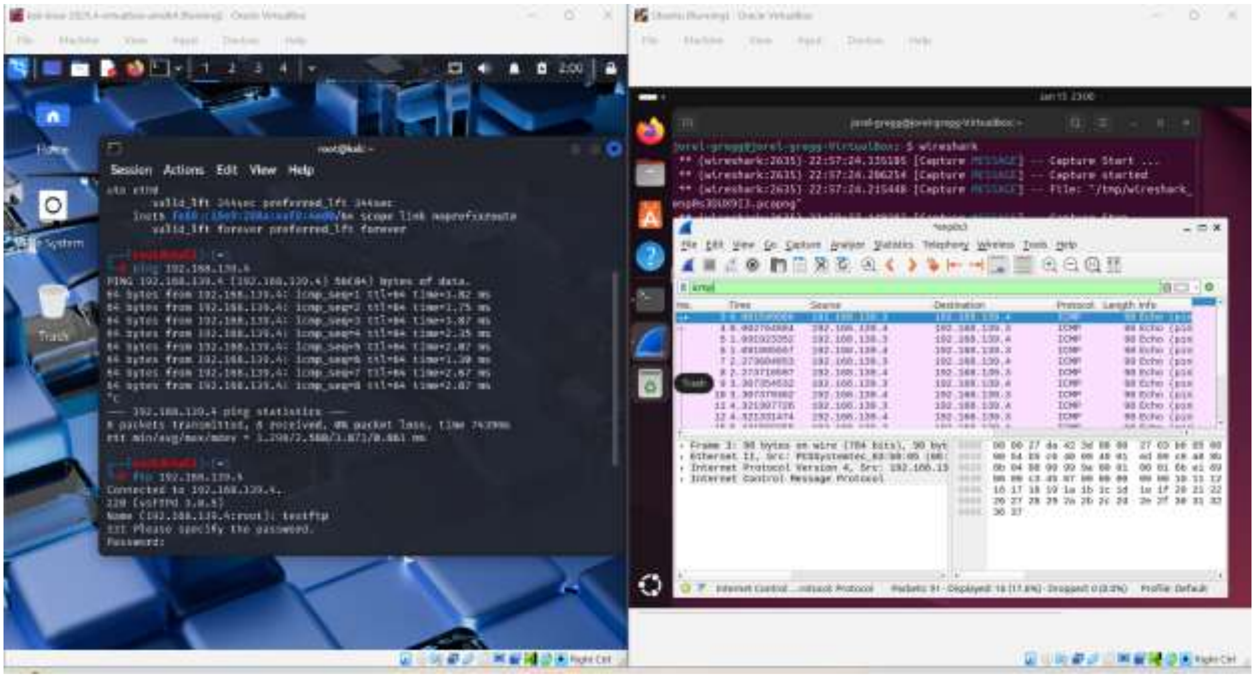
```
Capturing from enp0s3

No.  Time         Source                 Destination            Protocol  Length  Info
  1  0.000000000  192.168.139.4          192.168.139.2          DHCP         342  DHCP Request
  2  0.011400829  PCSSystemtec_3a:9f:…   Broadcast              ARP           60  Who has 192.16…
  3  0.011448129  PCSSystemtec_da:42:…   PCSSystemtec_3a:9f:…   ARP           42  192.168.139.4 i…
  4  0.011489375  192.168.139.2          192.168.139.4          DHCP         590  DHCP ACK

› Frame 1: 342 bytes on wire (2736 bits), 342
› Ethernet II, Src: PCSSystemtec_da:42:3d (08:
› Internet Protocol Version 4, Src: 192.168.13
› User Datagram Protocol, Src Port: 68, Dst Po
› Dynamic Host Configuration Protocol (Request

enp0s3: <live capture in progress>    Packets: 4 · Displayed: 4 (100.0%)    Profile: Default
```

4. Then I was able to start testing my first real traffic between the two VM's using a simple ping command from Kali-Linux to Ubuntu. We can filter by ICMP, and we can see the ping traffic which may point to someone trying to map the network and should be watched if continually from the same source.



```
uta eth0
       valid_lft 244sec preferred_lft 244sec
    inet6 fe80::1b69:208a:aaf2:4ed0/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

┌──(root㉿kali)-[~]
└─# ping 192.168.139.4
PING 192.168.139.4 (192.168.139.4) 56(84) bytes of data.
64 bytes from 192.168.139.4: icmp_seq=1 ttl=64 time=3.82 ms
64 bytes from 192.168.139.4: icmp_seq=2 ttl=64 time=1.75 ms
64 bytes from 192.168.139.4: icmp_seq=3 ttl=64 time=3.87 ms
64 bytes from 192.168.139.4: icmp_seq=4 ttl=64 time=2.35 ms
64 bytes from 192.168.139.4: icmp_seq=5 ttl=64 time=2.87 ms
64 bytes from 192.168.139.4: icmp_seq=6 ttl=64 time=1.30 ms
64 bytes from 192.168.139.4: icmp_seq=7 ttl=64 time=2.67 ms
64 bytes from 192.168.139.4: icmp_seq=8 ttl=64 time=2.87 ms
^C
--- 192.168.139.4 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7439ms
rtt min/avg/max/mdev = 1.298/2.588/3.871/0.861 ms

┌──(root㉿kali)-[~]
└─# ftp 192.168.139.4
Connected to 192.168.139.4.
220 (vsFTPd 3.0.5)
Name (192.168.139.4:root): testftp
331 Please specify the password.
Password:
```

```
jorel-gregg@jorel-gregg-VirtualBox:~$ wireshark
 ** (wireshark:2635) 22:57:24.335185 [Capture MESSAGE] -- Capture Start ...
 ** (wireshark:2635) 22:57:24.286254 [Capture MESSAGE] -- Capture started
 ** (wireshark:2635) 22:57:24.215448 [Capture MESSAGE] -- File: "/tmp/wireshark_enp0s3[illegible].pcapng"
```

Filter: icmp

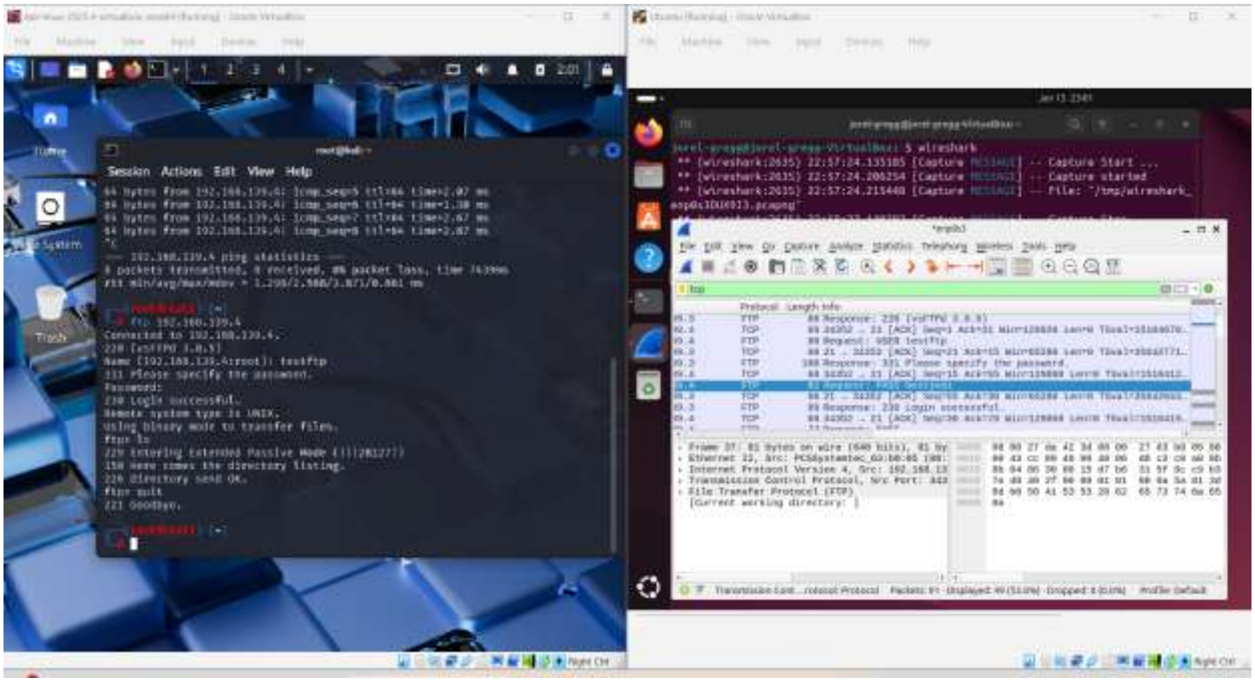
```
No.  Time          Source          Destination     Protocol Length Info
   3 0.001599009   192.168.139.3   192.168.139.4   ICMP     98 Echo (pin
   4 0.002764884   192.168.139.4   192.168.139.3   ICMP     98 Echo (pin
   5 1.001923252   192.168.139.3   192.168.139.4   ICMP     98 Echo (pin
   6 1.001886667   192.168.139.4   192.168.139.3   ICMP     98 Echo (pin
   7 2.273604853   192.168.139.3   192.168.139.4   ICMP     98 Echo (pin
   8 2.273710887   192.168.139.4   192.168.139.3   ICMP     98 Echo (pin
   9 3.307354532   192.168.139.3   192.168.139.4   ICMP     98 Echo (pin
  10 3.307378882   192.168.139.4   192.168.139.3   ICMP     98 Echo (pin
  11 4.321307726   192.168.139.3   192.168.139.4   ICMP     98 Echo (pin
  12 4.321331474   192.168.139.4   192.168.139.3   ICMP     98 Echo (pin
```

```
› Frame 3: 98 bytes on wire (784 bits), 98 byt
› Ethernet II, Src: PCSSystemtec_83:98:05 (08:
› Internet Protocol Version 4, Src: 192.168.13
› Internet Control Message Protocol
```

```
00 00 27 da 42 3d 08 00  27 c0 b0 05 08
00 54 e9 c0 40 00 40 01  ed 00 c0 a8 8b
8b 04 08 00 99 5a 00 01  00 01 6b e1 69
00 00 c3 49 07 00 00 00  00 00 10 11 12
16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22
26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32
36 37
```

Internet Control ...ntrol Protocol   Packets: 91 · Displayed: 16 (17.6%) · Dropped: 0 (0.0%)   Profile: Default

5. The next step was to intentionally set up and use a temporary FTP user for us to find them with Kali-Linux and capture the traffic with Wireshark. I did this by installing and running *VSFTPD* on the Ubuntu machine and then creating a new user with a password for the Kali machine to log into. This allowed me to capture and see FTP packets that showed the password in plain text which is why encryption is so important.



```
64 bytes from 192.168.139.4: icmp_seq=5 ttl=64 time=2.07 ms
64 bytes from 192.168.139.4: icmp_seq=6 ttl=64 time=1.38 ms
64 bytes from 192.168.139.4: icmp_seq=7 ttl=64 time=2.67 ms
64 bytes from 192.168.139.4: icmp_seq=8 ttl=64 time=2.87 ms
^C
--- 192.168.139.4 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7039ms
rtt min/avg/max/mdev = 1.298/2.508/3.871/0.881 ms

┌──(root㉿kali)-[~]
└─# ftp 192.168.139.4
Connected to 192.168.139.4.
220 (vsFTPd 3.0.5)
Name (192.168.139.4:root): testftp
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||28127|)
150 Here comes the directory listing.
226 Directory send OK.
ftp> quit
221 Goodbye.

┌──(root㉿kali)-[~]
└─#
```

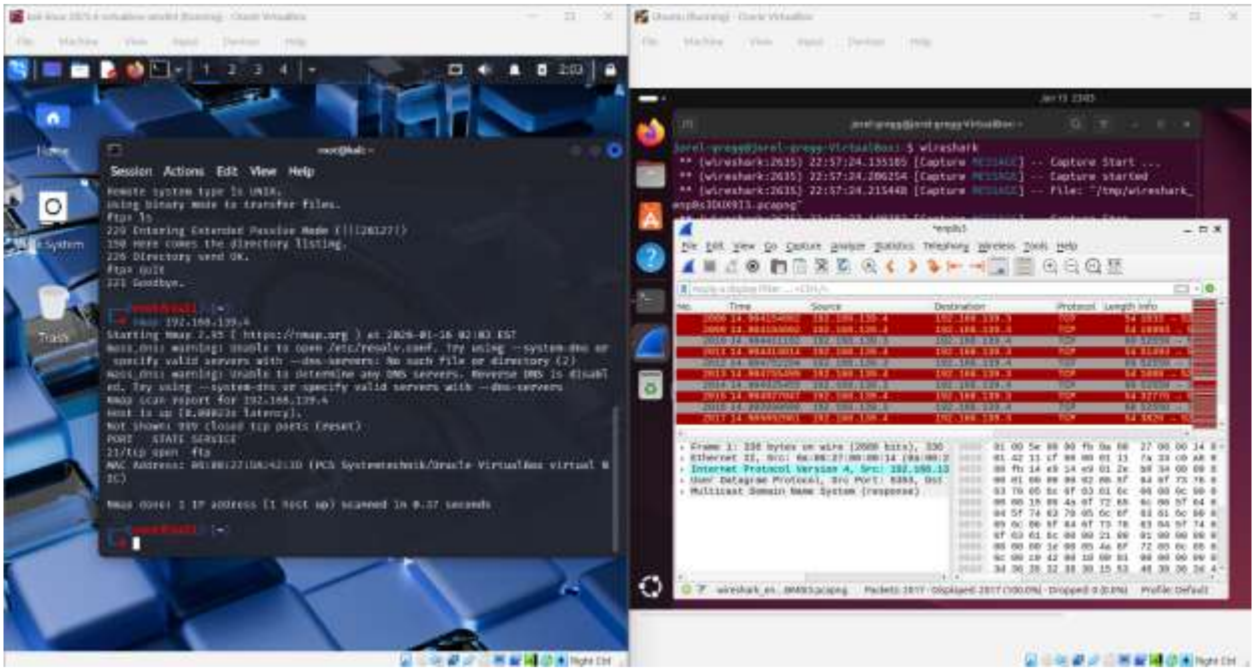
```
jerel-gregg@jerel-gregg-VirtualBox:~$ wireshark
 ** (wireshark:2635) 22:57:24.135185 [Capture MESSAGE] -- Capture Start ...
 ** (wireshark:2635) 22:57:24.206254 [Capture MESSAGE] -- Capture started
 ** (wireshark:2635) 22:57:24.215448 [Capture MESSAGE] -- File: "/tmp/wireshark_enp0s3DUX913.pcapng"
```

```
ftp
Protocol  Length Info
FTP           88 Response: 220 (vsFTPd 3.0.5)
TCP           66 34352 → 21 [ACK] Seq=1 Ack=21 Win=128896 Len=0 TSval=15168070…
FTP           80 Request: USER testftp
TCP           66 21 → 34352 [ACK] Seq=21 Ack=15 Win=65280 Len=0 TSval=35842771…
FTP          100 Response: 331 Please specify the password.
TCP           66 34352 → 21 [ACK] Seq=15 Ack=55 Win=128896 Len=0 TSval=15168012…
FTP           81 Request: PASS [illegible]
TCP           66 21 → 34352 [ACK] Seq=55 Ack=30 Win=65280 Len=0 TSval=35842822…
FTP           89 Response: 230 Login successful.
TCP           66 34352 → 21 [ACK] Seq=30 Ack=78 Win=128896 Len=0 TSval=15168018…

> Frame 37: 81 bytes on wire (648 bits), 81 by
> Ethernet II, Src: PCSSystemtec_63:b0:65 (08:
> Internet Protocol Version 4, Src: 192.168.13
> Transmission Control Protocol, Src Port: 343
> File Transfer Protocol (FTP)
  [Current working directory: ]
```

Session Actions Edit View Help
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Transmission Control Protocol   Packets: 91 · Displayed: 49 (53.8%) · Dropped: 0 (0.0%)   Profile: Default
Right Ctrl

6. I then used NMAP to scan the ports of the Ubuntu machine from the Kali machine and captured the TCP packets in Wireshark. Wireshark begins to flag these red when it notices many incoming packets testing different ports with the same source and destination IP addresses milliseconds apart. I then was able to filter them further to identify that they were SYN packets which were testing the connection without fully connecting, indicating scanning that is most likely a nefarious reconnaissance.



```
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||28127|)
150 Here comes the directory listing.
226 Directory send OK.
ftp> quit
221 Goodbye.

  nmap 192.168.139.4
Starting Nmap 7.95 ( https://nmap.org ) at 2026-01-16 02:03 EST
mass_dns: warning: Unable to open /etc/resolv.conf. Try using --system-dns or
 specify valid servers with --dns-servers: No such file or directory (2)
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabl
ed. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.139.4
Host is up (0.00023s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
21/tcp open  ftp
MAC Address: 08:00:27:D8:42:3D (PCS Systemtechnik/Oracle VirtualBox virtual N
IC)

Nmap done: 1 IP address (1 host up) scanned in 0.37 seconds
```

```
jorel-gregg@jorel-gregg-VirtualBox:~$ wireshark
 ** (wireshark:2635) 22:57:24.135185 [Capture MESSAGE] -- Capture Start ...
 ** (wireshark:2635) 22:57:24.206254 [Capture MESSAGE] -- Capture started
 ** (wireshark:2635) 22:57:24.215448 [Capture MESSAGE] -- File: "/tmp/wireshark_
enp0s3DUX9T3.pcapng"
```

Session Actions Edit View Help
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
No. Time Source Destination Protocol Length Info