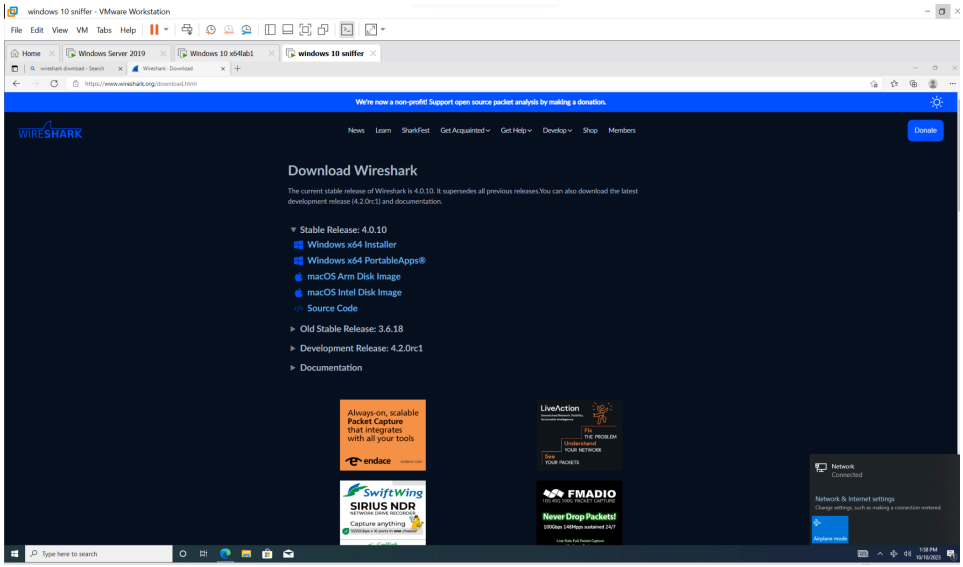
Julianna Russo
Network Security
Professor Cannistra
Lab 02- Analyzing Network Traffic Using Packet Capture Software

Description:

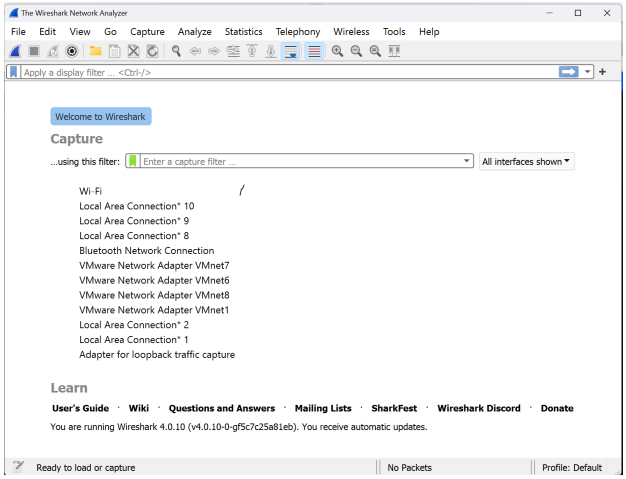
The purpose of this lab is to become more efficient at capturing and analyzing network traffic using packet capture software. Sometimes this is done by using automated tools and other times manual methods are used to accomplish this. This lab uses Wireshark as its packet-capturing software.

Task One:

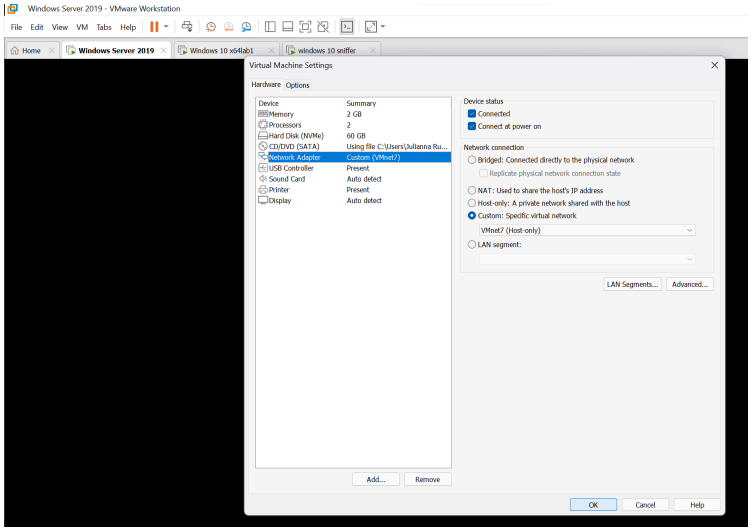
Create a sniffer server and connect to the internet

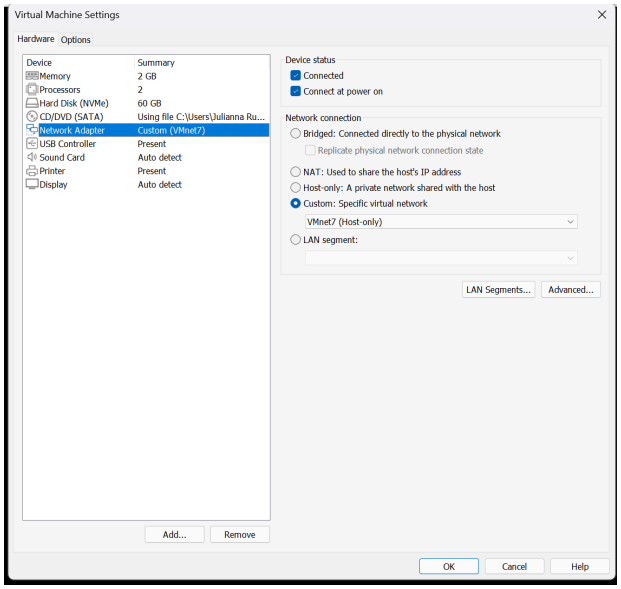


Install Wireshark on Host Machine

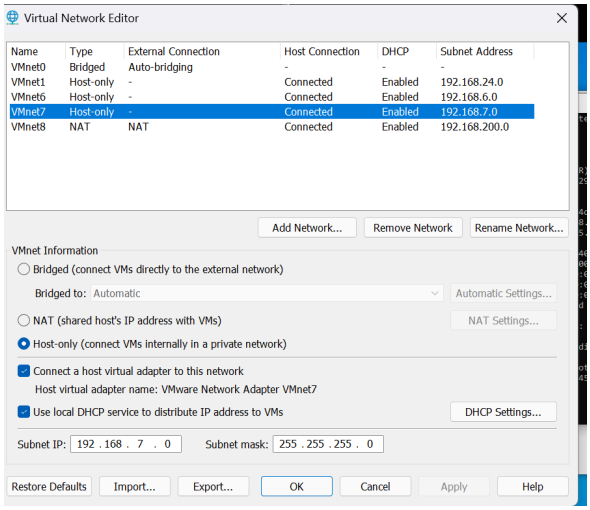


Isolate VM server and client

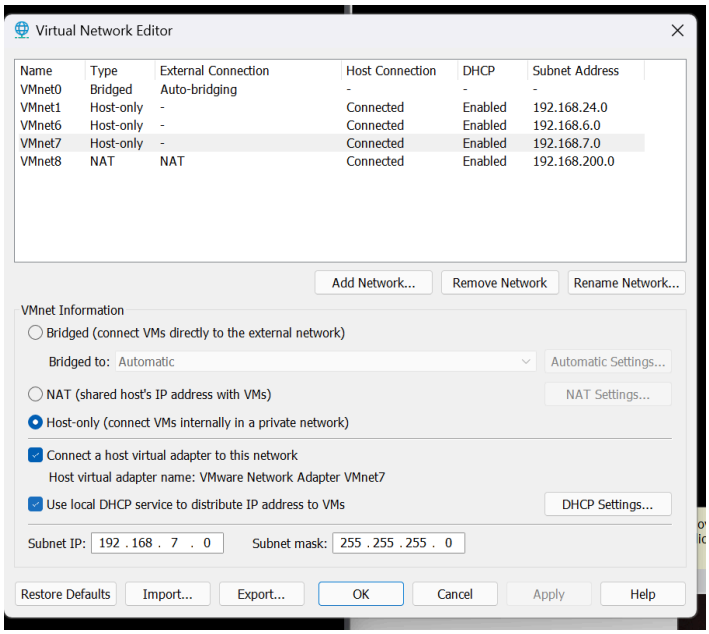




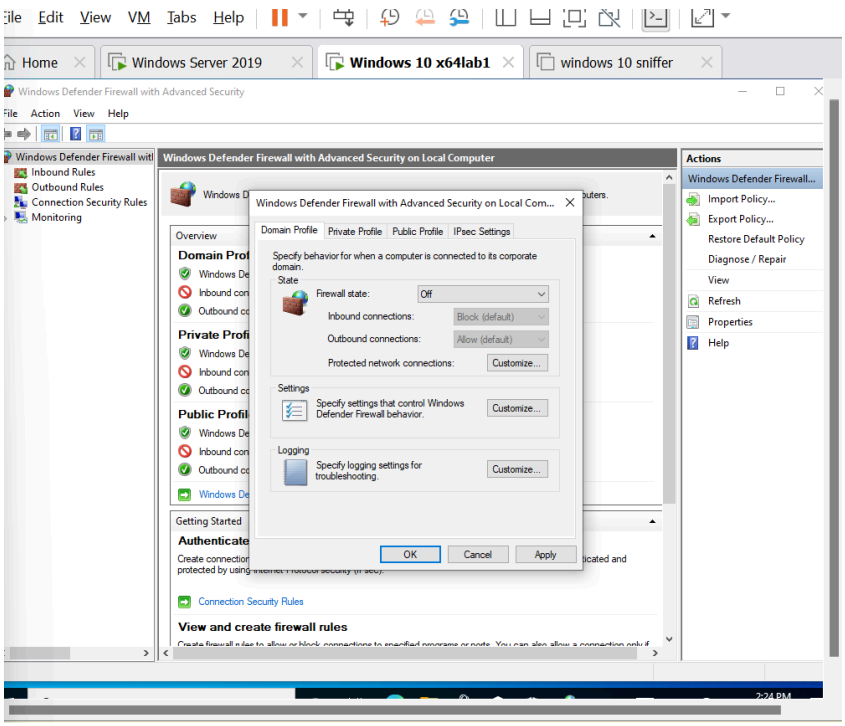
Windows 10



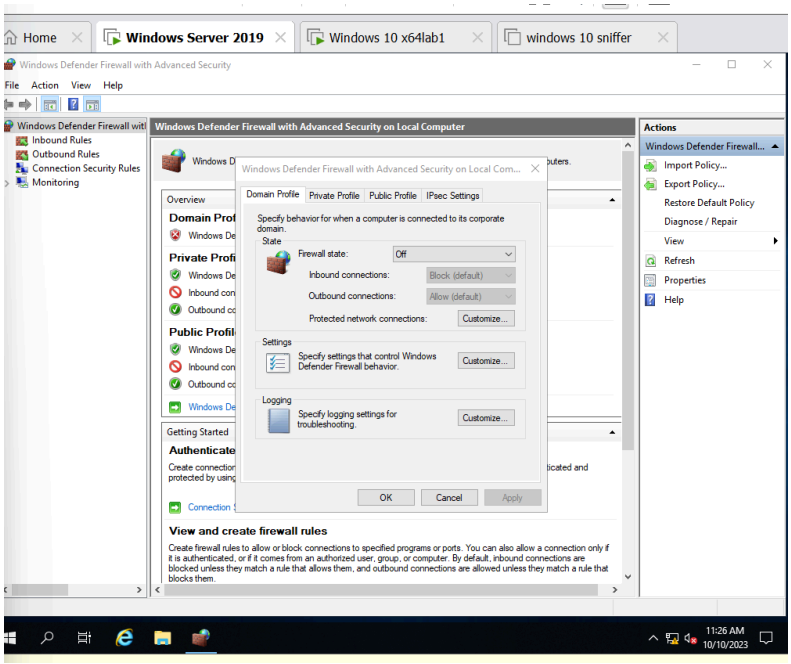
Windows 2019 server



Disable Firewall
Windows 10

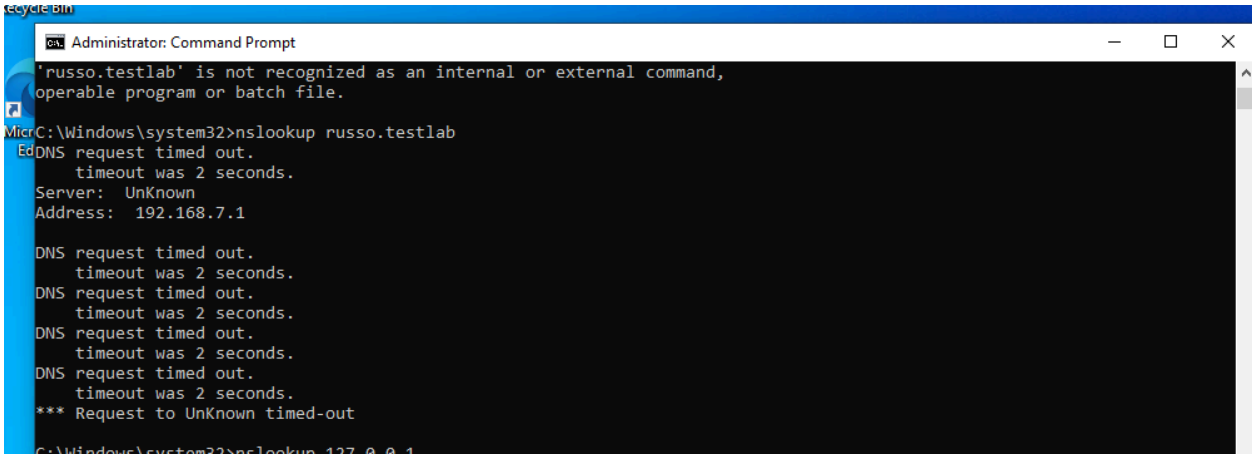


Windows 2019 server



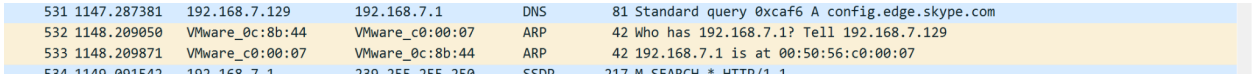
Task 2

A)



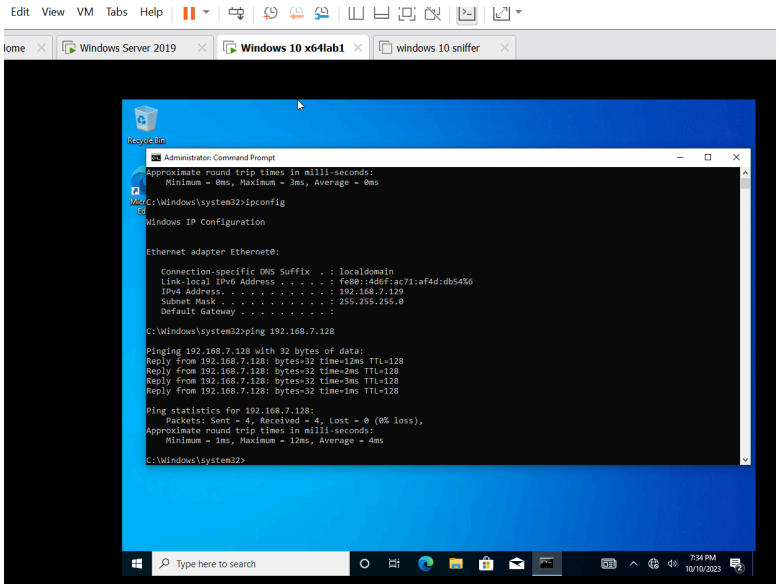
This is the command prompt for the DNS lookup by name on the client to the server

This is the result from Wireshark, This shows the traffic caused by the nslookup command. The yellow sections of the screenshot are what is going on during the ping command, the server is looking for a certain ip address, and giving to a destination address. The ping command is very useful in network security, or even just in computer science as it is a great way to test and establish for connectivity.



531	1147.287381	192.168.7.129	192.168.7.1	DNS	81 Standard query 0xcaf6 A config.edge.skype.com
532	1148.209050	VMware_0c:8b:44	VMware_c0:00:07	ARP	42 Who has 192.168.7.1? Tell 192.168.7.129
533	1148.209871	VMware_c0:00:07	VMware_0c:8b:44	ARP	42 192.168.7.1 is at 00:50:56:c0:00:07

Pinging windows server from windows client



Ping successful.

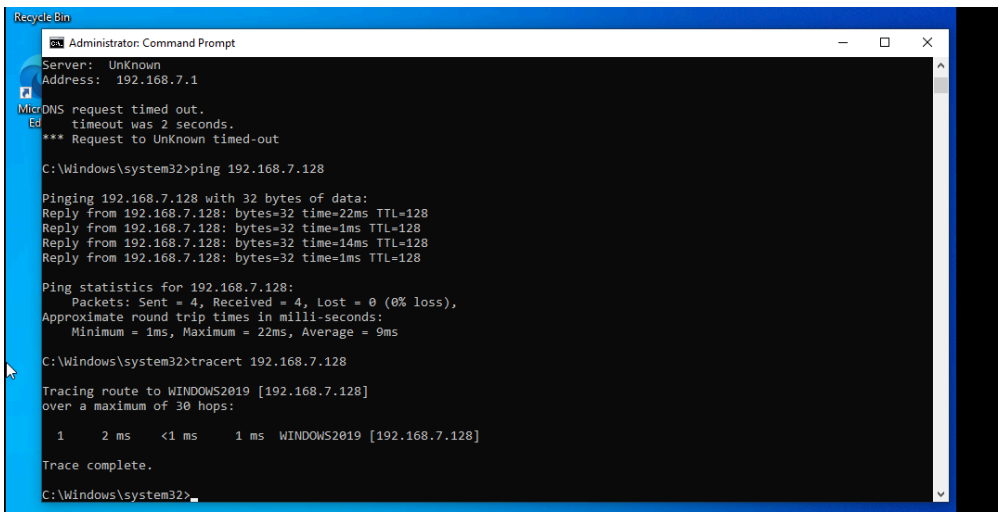
This is the result from Wireshark

No.	Time	Source	Destination	Protocol	Length	Info
548	1245.865445	192.168.7.129	192.168.7.1	DNS	78	Standard query 0x934f A wpad.russo.testlab
549	1257.108645	192.168.7.129	192.168.7.128	ICMP	74	Echo (ping) request id=0x0001, seq=9/2304, ttl=128 (reply in 550)
550	1257.125652	192.168.7.128	192.168.7.129	ICMP	74	Echo (ping) reply id=0x0001, seq=9/2304, ttl=128 (request in 549)
551	1258.136141	192.168.7.129	192.168.7.128	ICMP	74	Echo (ping) request id=0x0001, seq=10/2560, ttl=128 (reply in 552)
552	1258.136768	192.168.7.128	192.168.7.129	ICMP	74	Echo (ping) reply id=0x0001, seq=10/2560, ttl=128 (request in 551)
553	1259.163031	192.168.7.129	192.168.7.128	ICMP	74	Echo (ping) request id=0x0001, seq=11/2816, ttl=128 (reply in 554)
554	1259.163649	192.168.7.128	192.168.7.129	ICMP	74	Echo (ping) reply id=0x0001, seq=11/2816, ttl=128 (request in 553)
555	1260.193724	192.168.7.129	192.168.7.128	ICMP	74	Echo (ping) request id=0x0001, seq=12/3072, ttl=128 (reply in 556)
556	1260.194505	192.168.7.128	192.168.7.129	ICMP	74	Echo (ping) reply id=0x0001, seq=12/3072, ttl=128 (request in 555)
557	1261.708681	VMware_0c:8b:44	VMware_fe:02:b4	ARP	42	Who has 192.168.7.128? Tell 192.168.7.129
558	1261.710160	VMware_fe:02:b4	VMware_0c:8b:44	ARP	42	192.168.7.128 is at 00:0c:29:fe:02:b4
559	1262.128939	VMware_fe:02:b4	VMware_0c:8b:44	ARP	42	Who has 192.168.7.129? Tell 192.168.7.128
560	1262.129459	VMware_0c:8b:44	VMware_fe:02:b4	ARP	42	192.168.7.129 is at 00:0c:29:0c:8b:44

B) This is the tracert command used to trace the packets going from th windows client to the windows server

In the command prompt window, the tracert command will report the number of hops to get to the desired server. There was one hop, and it took less than 1 ms to complete. In wireshark the results look similar to the ping command results.

768	1697.208807	VMware_0c:8b:44	VMware_c0:00:07	ARP	42	Who has 192.168.7.1? Tell 192.168.7.129
769	1697.209626	VMware_c0:00:07	VMware_0c:8b:44	ARP	42	192.168.7.1 is at 00:50:56:c0:00:07
770	1699.568692	192.168.7.129	192.168.7.1	DNS	91	Standard query 0xd446 A settings-win.data.microsoft.com
771	1701.053485	192.168.7.129	192.168.7.1	DNS	85	Standard query 0xc018 A localdomain.russo.testlab
772	1701.053572	192.168.7.129	192.168.7.1	DNS	86	Standard query 0xd09c A slscr.update.microsoft.com



C) This is the arp cache table and ipconfig2. on windows client

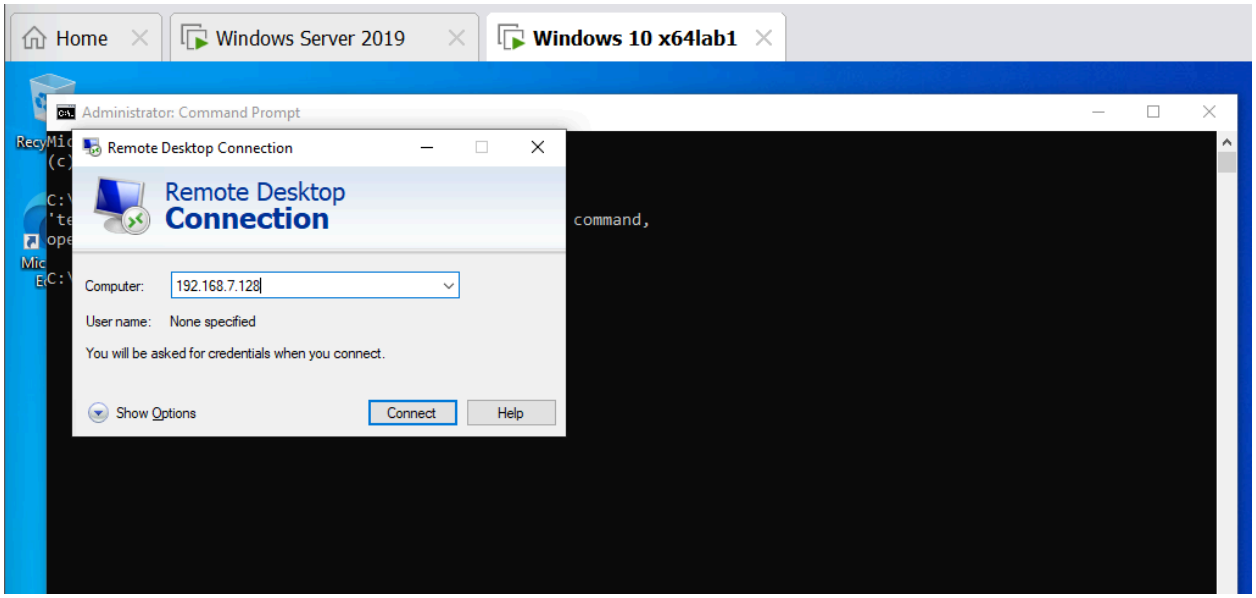
The arp cache table is a mapping table between IP addresses and MAC addresses. It is used by hosts and routers to store the MAC addresses of devices that are directly or indirectly attached to the network.

```
:\Windows\system32>arp /a

Interface: 192.168.7.129 --- 0x6
  Internet Address      Physical Address      Type
  192.168.7.1           00-50-56-c0-00-07     dynamic
  192.168.7.128         00-0c-29-fe-02-b4     dynamic
  192.168.7.254         00-50-56-ef-6a-8f     dynamic
  192.168.7.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

:\Windows\system32>\ipconfig
'\ipconfig' is not recognized as an internal or external command,
perable program or batch file.

:\Windows\system32>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . : localdomain
   Link-local IPv6 Address . . . . . : fe80::4d6f:ac71:af4d:db54%6
   IPv4 Address. . . . . . . . . . . : 192.168.7.129
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

:\Windows\system32>_
```

This is it on the windows 2019 server

```
C:\Users\Administrator>arp /a

Interface: 192.168.7.128 --- 0x9
  Internet Address      Physical Address      Type
  192.168.7.129         00-0c-29-0c-8b-44     dynamic
  192.168.7.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static

C:\Users\Administrator>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::e535:a2a7:5ee2:4b6a%9
   IPv4 Address. . . . . . . . . . . : 192.168.7.128
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :
```

D) This is the remote desktop connection on the windows 10 client that goes to the windows server. I had to login to the server using the administrator credentials



No.	Time	Source	Destination	Protocol	Length	Info
8363	14194.208791	192.168.7.129	192.168.7.1	DNS	125	Standard query 0x53d2 SRV _ldap._tcp.Default-First-Site-Name._sites.dc._
8364	14195.146229	192.168.7.129	239.255.255.250	SSDP	179	M-SEARCH * HTTP/1.1
8365	14196.177213	192.168.7.129	192.168.7.1	DNS	83	Standard query 0x4f09 SOA Windows10.russo.testlab
8366	14196.224890	192.168.7.129	192.168.7.1	DNS	125	Standard query 0x53d2 SRV _ldap._tcp.Default-First-Site-Name._sites.dc._
8367	14196.958564	VMware_0c:8b:44	VMware_ef:6a:8f	ARP	42	Who has 192.168.7.254? Tell 192.168.7.129
8368	14196.958882	VMware_ef:6a:8f	VMware_0c:8b:44	ARP	42	192.168.7.254 is at 00:50:56:ef:6a:8f
8369	14196.958924	VMware_0c:8b:44	VMware_c0:00:07	ARP	42	Who has 192.168.7.1? Tell 192.168.7.129
8370	14196.959506	VMware_c0:00:07	VMware_0c:8b:44	ARP	42	192.168.7.1 is at 00:50:56:c0:00:07
8371	14198.146628	192.168.7.129	239.255.255.250	SSDP	179	M-SEARCH * HTTP/1.1
8372	14199.763779	192.168.7.129	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
8373	14200.192759	192.168.7.129	192.168.7.1	DNS	83	Standard query 0x4f09 SOA Windows10.russo.testlab
8374	14200.240250	192.168.7.129	192.168.7.1	DNS	125	Standard query 0x53d2 SRV _ldap._tcp.Default-First-Site-Name._sites.dc._
8375	14200.771133	192.168.7.129	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
8376	14201.176206	192.168.7.129	239.255.255.250	SSDP	179	M-SEARCH * HTTP/1.1
8377	14201.771784	192.168.7.129	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
8378	14202.787314	192.168.7.129	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
8379	14204.161889	192.168.7.129	239.255.255.250	SSDP	179	M-SEARCH * HTTP/1.1
8380	14204.259022	192.168.7.129	192.168.7.1	DNS	85	Standard query 0xdc35 SRV localdomain.russo.testlab
8381	14205.256177	192.168.7.129	192.168.7.1	DNS	85	Standard query 0xdc35 SRV localdomain.russo.testlab
8382	14206.271309	192.168.7.129	192.168.7.1	DNS	85	Standard query 0xdc35 SRV localdomain.russo.testlab
8383	14207.177259	192.168.7.129	239.255.255.250	SSDP	179	M-SEARCH * HTTP/1.1
8384	14208.271730	192.168.7.129	192.168.7.1	DNS	85	Standard query 0xdc35 SRV localdomain.russo.testlab
8385	14212.271849	192.168.7.129	192.168.7.1	DNS	85	Standard query 0xdc35 SRV localdomain.russo.testlab
8386	14216.289394	192.168.7.129	192.168.7.1	DNS	85	Standard query 0x1769 SRV localdomain.russo.testlab
8387	14217.302190	192.168.7.129	192.168.7.1	DNS	85	Standard query 0x1769 SRV localdomain.russo.testlab
8388	14218.302548	192.168.7.129	192.168.7.1	DNS	85	Standard query 0x1769 SRV localdomain.russo.testlab
8389	14220.318666	192.168.7.129	192.168.7.1	DNS	85	Standard query 0x1769 SRV localdomain.russo.testlab
8390	14224.333659	192.168.7.129	192.168.7.1	DNS	85	Standard query 0x1769 SRV localdomain.russo.testlab

E)

This is the windows 10 client net use command

The net use command connects/configures connections to shared resources like mapped drives. The wireshark screenshot shows this by the create lines in the info column.These results aren’t surprising.

```
C:\Users\testuser1>net use Z: \\192.168.7.128\C$ /user:192.168.7.128\Administrator
Enter the password for '192.168.7.128\Administrator' to connect to '192.168.7.128':
The command completed successfully.

C:\Users\testuser1>
```

No.	Time	Source	Destination	Protocol	Length	Info
7130	11356.326506	192.168.7.128	192.168.7.129	TCP	54	445 → 49677 [ACK] Seq=1519 Ack=1549 Win=2102272 Len=0
7131	11356.341410	192.168.7.128	192.168.7.129	SMB2	130	Ioctl Response, Error: STATUS_PENDING
7132	11356.343198	192.168.7.129	192.168.7.128	TCP	54	49677 → 445 [ACK] Seq=1549 Ack=1595 Win=262656 Len=0
7133	11356.628191	192.168.7.128	192.168.7.129	SMB2	130	Ioctl Response, Error: STATUS_NOT_FOUND
7134	11356.637265	192.168.7.129	192.168.7.128	SMB2	166	Tree Connect Request Tree: \\192.168.7.128\C$
7135	11356.639266	192.168.7.128	192.168.7.129	SMB2	138	Tree Connect Response
7136	11356.685276	192.168.7.128	192.168.7.129	TCP	138	[TCP Retransmission] 445 → 49677 [PSH, ACK] Seq=1671 Ack=1661 Win=2102272…
7137	11356.685627	192.168.7.129	192.168.7.128	TCP	66	49677 → 445 [ACK] Seq=1661 Ack=1755 Win=262400 Len=0 SLE=1671 SRE=1755
7138	11356.881817	192.168.7.129	192.168.7.128	SMB2	382	Create Request File: Desktop.ini
7139	11356.884199	192.168.7.128	192.168.7.129	SMB2	130	Create Response, Error: STATUS_OBJECT_NAME_NOT_FOUND
7140	11356.894000	192.168.7.129	192.168.7.128	SMB2	250	Create Request File: AutoRun.inf
7141	11356.895352	192.168.7.128	192.168.7.129	SMB2	130	Create Response, Error: STATUS_OBJECT_NAME_NOT_FOUND
7142	11356.935085	192.168.7.128	192.168.7.129	TCP	130	[TCP Retransmission] 445 → 49677 [PSH, ACK] Seq=1831 Ack=2185 Win=2101760…
7143	11356.935528	192.168.7.129	192.168.7.128	TCP	66	49677 → 445 [ACK] Seq=2185 Ack=1907 Win=262400 Len=0 SLE=1831 SRE=1907
7144	11357.419364	192.168.7.129	192.168.7.128	SMB2	234	Create Request File:
7145	11357.420929	192.168.7.128	192.168.7.129	SMB2	298	Create Response File:
7146	11357.426332	192.168.7.129	192.168.7.128	SMB2	146	Close Request File:
7147	11357.428109	192.168.7.128	192.168.7.129	SMB2	182	Close Response
7148	11357.466504	192.168.7.128	192.168.7.129	TCP	182	[TCP Retransmission] 445 → 49677 [PSH, ACK] Seq=2151 Ack=2457 Win=2101248…
7149	11357.467154	192.168.7.129	192.168.7.128	TCP	66	49677 → 445 [ACK] Seq=2457 Ack=2279 Win=261888 Len=0 SLE=2151 SRE=2279
7150	11366.936618	192.168.7.129	192.168.7.128	SMB2	126	Tree Disconnect Request
7151	11366.937577	192.168.7.128	192.168.7.129	SMB2	126	Tree Disconnect Response
7152	11366.981530	192.168.7.128	192.168.7.129	TCP	126	[TCP Retransmission] 445 → 49677 [PSH, ACK] Seq=2279 Ack=2529 Win=2101248…
7153	11366.982276	192.168.7.129	192.168.7.128	TCP	66	49677 → 445 [ACK] Seq=2529 Ack=2351 Win=261888 Len=0 SLE=2279 SRE=2351
7154	11396.953294	192.168.7.129	192.168.7.128	TCP	55	[TCP Keep-Alive] 49677 → 445 [ACK] Seq=2528 Ack=2351 Win=261888 Len=1
7155	11396.953807	192.168.7.128	192.168.7.129	TCP	66	[TCP Keep-Alive ACK] 445 → 49677 [ACK] Seq=2351 Ack=2529 Win=2101248 Len=…
7156	11401.623699	VMware_fe:02:b4	VMware_0c:8b:44	ARP	42	Who has 192.168.7.129? Tell 192.168.7.128
7157	11401.624437	VMware_0c:8b:44	VMware_fe:02:b4	ARP	42	192.168.7.129 is at 00:0c:29:0c:8b:44

F) This is the result from the ping on windows client to the windows 2019 server
The ping is pretty standard as far as wireshark goes, it has the call and response.

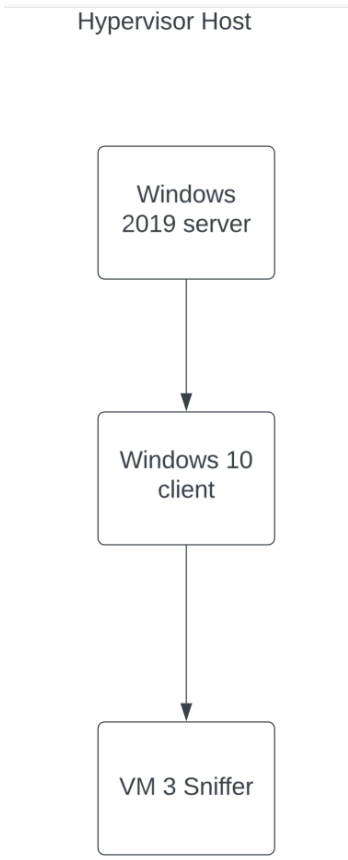
Apply a display filter … <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
7309	11657.387127	192.168.7.129	192.168.7.1	DNS	78	Standard query 0x6c01 A wpad.russo.testlab
7310	11657.965321	VMware_0c:8b:44	VMware_c0:00:07	ARP	42	Who has 192.168.7.1? Tell 192.168.7.129
7311	11657.965812	VMware_c0:00:07	VMware_0c:8b:44	ARP	42	192.168.7.1 is at 00:50:56:c0:00:07
7312	11661.402490	192.168.7.129	192.168.7.1	DNS	78	Standard query 0x6c01 A wpad.russo.testlab
7313	11664.917019	192.168.7.1	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
7314	11665.919213	192.168.7.1	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
7315	11666.919701	192.168.7.1	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
7316	11667.920177	192.168.7.1	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
7317	11669.033067	192.168.7.128	192.168.7.255	BROWSER	255	Host Announcement WINDOWS2019, Workstation, Server, Domain Controller, Ti…
7318	11679.530392	192.168.7.129	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
7319	11680.544022	192.168.7.129	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
7320	11681.559197	192.168.7.129	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
7321	11682.575609	192.168.7.129	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
7322	11738.060462	192.168.7.128	192.168.7.129	TCP	55	[TCP Keep-Alive] 445 → 49677 [ACK] Seq=2770 Ack=2653 Win=2101248 Len=1
7323	11738.062148	192.168.7.129	192.168.7.128	TCP	66	[TCP Keep-Alive ACK] 49677 → 445 [ACK] Seq=2653 Ack=2771 Win=261376 Len=0…
7324	11742.965159	VMware_0c:8b:44	VMware_fe:02:b4	ARP	42	Who has 192.168.7.128? Tell 192.168.7.129
7325	11742.966230	VMware_fe:02:b4	VMware_0c:8b:44	ARP	42	192.168.7.128 is at 00:0c:29:fe:02:b4
7326	11768.532362	192.168.7.129	192.168.7.128	ICMP	74	Echo (ping) request id=0x0001, seq=1/256, ttl=128 (reply in 7327)
7327	11768.535005	192.168.7.128	192.168.7.129	ICMP	74	Echo (ping) reply id=0x0001, seq=1/256, ttl=128 (request in 7326)
7328	11769.551950	192.168.7.129	192.168.7.128	ICMP	74	Echo (ping) request id=0x0001, seq=2/512, ttl=128 (reply in 7329)
7329	11769.552793	192.168.7.128	192.168.7.129	ICMP	74	Echo (ping) reply id=0x0001, seq=2/512, ttl=128 (request in 7328)
7330	11770.574640	192.168.7.129	192.168.7.128	ICMP	74	Echo (ping) request id=0x0001, seq=3/768, ttl=128 (reply in 7331)
7331	11770.575291	192.168.7.128	192.168.7.129	ICMP	74	Echo (ping) reply id=0x0001, seq=3/768, ttl=128 (request in 7330)
7332	11771.605910	192.168.7.129	192.168.7.128	ICMP	74	Echo (ping) request id=0x0001, seq=4/1024, ttl=128 (reply in 7333)
7333	11771.606766	192.168.7.128	192.168.7.129	ICMP	74	Echo (ping) reply id=0x0001, seq=4/1024, ttl=128 (request in 7332)
7334	11773.123051	VMware_fe:02:b4	VMware_0c:8b:44	ARP	42	Who has 192.168.7.129? Tell 192.168.7.128
7335	11773.126766	VMware_0c:8b:44	VMware_fe:02:b4	ARP	42	192.168.7.129 is at 00:0c:29:0c:8b:44
7336	11784.928165	192.168.7.1	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1

When I ran an ftp command this came up on wire shark, which is pretty interesting, but it is because the connection was refused.
This refusal was because i did not have the required data for the request. But what stands out to me is the colors used to indicate the refusal/error. The arrays of reds and dark colors would make it really easy to see something is wrong, quickly. This is a really intuitive design, and it makes recognition of problems quicker than if they used a more mundane color like blue or green.

7909	13092.543666	192.168.7.129	192.168.7.1	DNS	81	Standard query 0x5096 A config.edge.skype.com
7910	13096.556736	192.168.7.129	192.168.7.128	TCP	66	49682 → 21 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=1 SACK_PERM
7911	13096.558002	192.168.7.128	192.168.7.129	TCP	54	21 → 49682 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
7912	13097.060662	192.168.7.129	192.168.7.128	TCP	66	[TCP Retransmission] 49682 → 21 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=1 …
7913	13097.061429	192.168.7.128	192.168.7.129	TCP	54	21 → 49682 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
7914	13097.574724	192.168.7.129	192.168.7.128	TCP	66	[TCP Retransmission] 49682 → 21 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=1 …
7915	13097.575365	192.168.7.128	192.168.7.129	TCP	54	21 → 49682 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
7916	13098.089882	192.168.7.129	192.168.7.128	TCP	66	[TCP Retransmission] 49682 → 21 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=1 …
7917	13098.099298	192.168.7.128	192.168.7.129	TCP	54	21 → 49682 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
7918	13098.605899	192.168.7.129	192.168.7.128	TCP	66	[TCP Retransmission] 49682 → 21 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=1 …
7919	13098.606624	192.168.7.128	192.168.7.129	TCP	54	21 → 49682 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
7920	13101.122921	VMware_fe:02:b4	VMware_0c:8b:44	ARP	42	Who has 192.168.7.129? Tell 192.168.7.128
7921	13101.132016	VMware_0c:8b:44	VMware_fe:02:b4	ARP	42	192.168.7.129 is at 00:0c:29:0c:8b:44
7922	13101.465628	VMware_0c:8b:44	VMware_fe:02:b4	ARP	42	Who has 192.168.7.128? Tell 192.168.7.129
7923	13101.466737	VMware_fe:02:b4	VMware_0c:8b:44	ARP	42	192.168.7.128 is at 00:0c:29:fe:02:b4
7924	13105.002394	192.168.7.1	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
7925	13106.011611	192.168.7.1	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
7926	13107.013046	192.168.7.1	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
7927	13108.028550	192.168.7.1	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
7928	13110.526751	192.168.7.128	192.168.7.255	BROWSER	255	Host Announcement WINDOWS2019, Workstation, Server, Domain Controller, Ti…
7929	13115.960548	192.168.7.129	192.168.7.1	DNS	78	Standard query 0xfa48 A wpad.russo.testlab
7930	13116.949785	192.168.7.129	192.168.7.1	DNS	78	Standard query 0xfa48 A wpad.russo.testlab
7931	13117.964670	192.168.7.129	192.168.7.1	DNS	78	Standard query 0xfa48 A wpad.russo.testlab
7932	13119.682100	192.168.7.129	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
7933	13119.965314	192.168.7.129	192.168.7.1	DNS	78	Standard query 0xfa48 A wpad.russo.testlab
7934	13120.684219	192.168.7.129	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
7935	13121.699940	192.168.7.129	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1

Topology



Windows 2019	IPv4: 192.168.7.128	Subnet mask:255.255.255.0	MAC address: 00-0C-29-FE-02-B4	Credential Used: Username: Administrator

Windows 10	IPV4: 192.168.7.129	Subnet mask:255.255.255.0	MAC address: 00-0C-29-0C-88-44	Credential used username: win10_user

Syntax:

- Arp /a: displays a list of IP addresses and their corresponding MAC addresses for devices on your local network.
- Ipconfig : command that provides a simple description of your computer's network configuration, including details such as your IP address, subnet mask, gateway, and more.
- Nslookup : `nslookup` used for querying and resolving domain name system (DNS) information, such as IP addresses associated with domain names or vice versa. It's often used to troubleshoot or verify DNS-related information.
- Net use Z:\\192.168.2.128\C$/user:192.168.7.128\Administrator : used to map a network drive (drive letter Z) on your local machine to a shared folder (C$) on a remote computer with the IP address 192.168.2.128. It connects using the username "Administrator" on the remote computer, which has the IP address 192.168.7.128.

Conclusion:

The purpose of this lab is to become more efficient at capturing and analyzing network traffic using packet capture software. Sometimes this is done by using automated tools and other times manual methods are used to accomplish this. This lab uses Wireshark as its packet-capturing software. I be;lieve that i did everything correctly, but when starting this lab I had trouble with it, and had to completely do it over. What I struggled with the most in this lab is understanding what to do when, but as we use VMWare more, I am feeling more confident in my abilities.

References:

Crystal. “How to Connect One Virtual Machine to Another.” *Best Backup, Recovery, Clone Software for Any Devices and Everyone*, AOMEI Backupper Utility, 8 Mar. 2023, www.ubackup.com/enterprise-backup/how-to-connect-one-virtual-machine-to-another.html#:~:text=How%20to%20connect%20two%20virtual%20machines%20in%20VMware,create%20a%20network%20adapter%20in%20VMware%20Workstation.%20.

How to Connect One Virtual Machine to Another (ubackup.com)