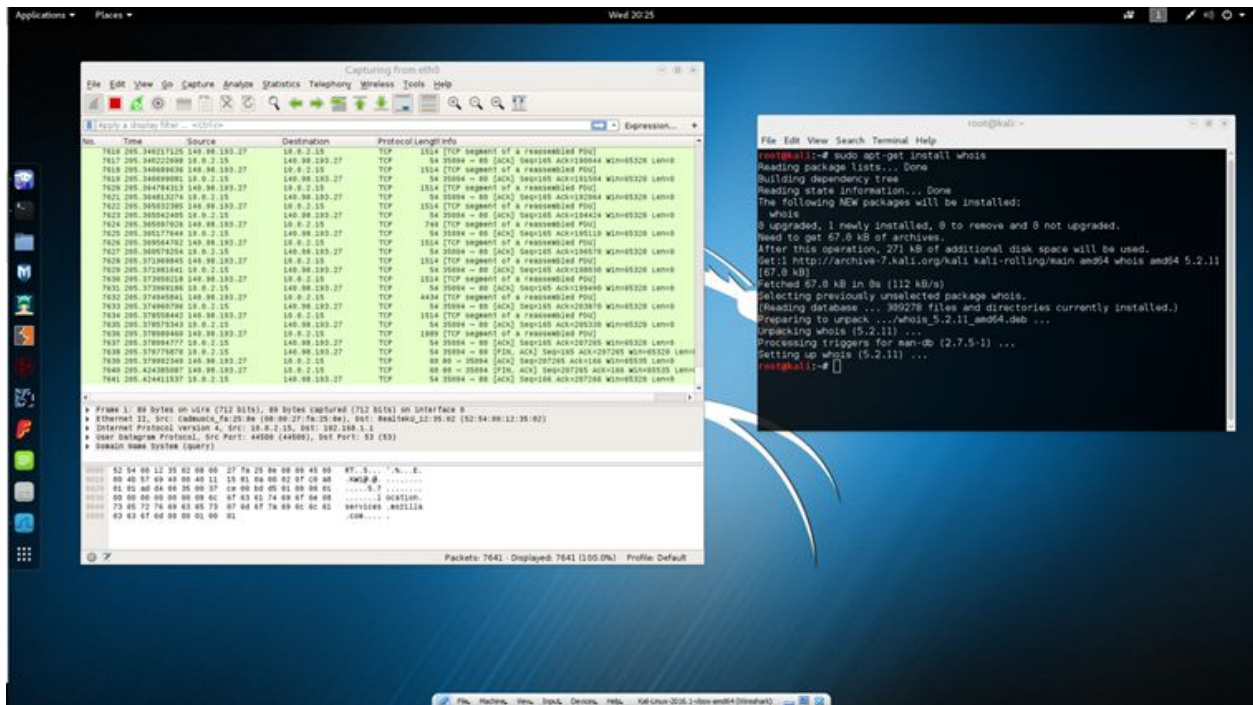


## Set up Kali Linux (2 Points)

Insert screenshot here [1 point]



- Explain the utility of the tool you've selected in the above screenshot. [1 point]

The tool in the screenshot above is called Wireshark. Wireshark is an open source packet analyzer, is written in C and C++. In this picture, I am using it to monitor the packets sent over the network connection of my Kali virtual machine.

## DNS Architecture (4 Points)

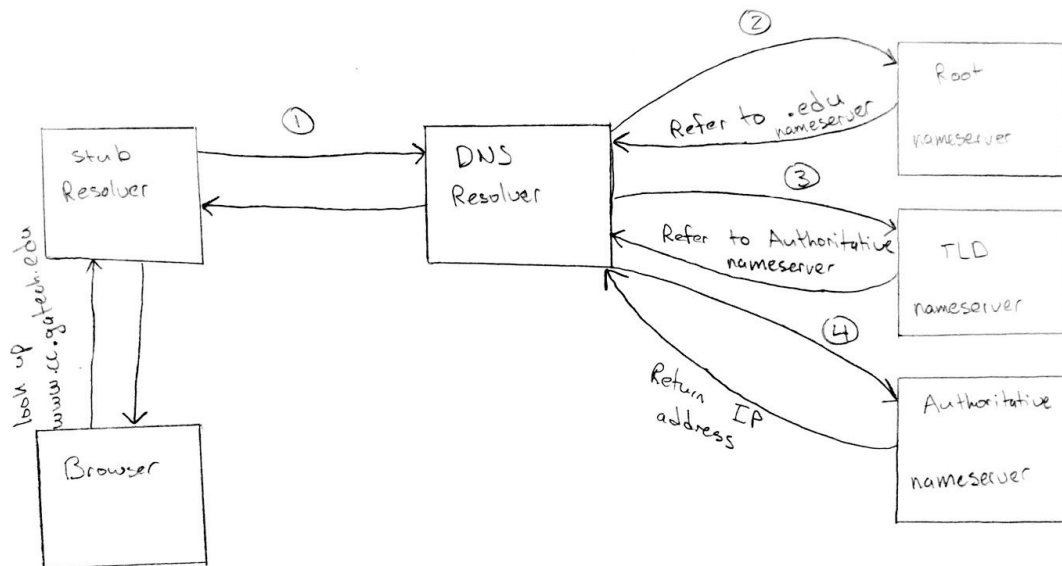
- In a DNS packet, what is the Transaction ID and how is it used? [0.5 points]

The transaction ID is a 16-bit field identifying a specific transaction, created by the message originator and copied by the responder into its response message. It is used for the DNS client to match responses to requests. (From technet.microsoft.com)

- What OSI layer four protocol does DNS use? Why? [0.5 points]

DNS primarily uses UDP over port 53, but may also use TCP in special circumstances. The UDP protocol is faster than TCP and therefore is advantageous for DNS because DNS messages are simple and require little overhead. TCP is used when the DNS responses exceed 512 bytes or when transferring data that cannot have any inconsistency such as Zone files.

- Draw a diagram showing the full trip for a recursive DNS query for `www.cc.gatech.edu`. Label each entity involved. (stub resolver, root server, etc.). Assume that nothing is cached. [1 point]



- Using the terminal of your Kali server, type “`whois gatech.edu`” Include the output in your submission. Now type “`traceroute www.cc.gatech.edu`” How could this information be useful to an attacker? (Note that if you try `traceroute` from the Kali system with NAT networking, the results may not be interesting. Use bridged networking or run this command on a physical system.) [0.5 points]

Domain Name: GATECH.EDU

Registrant:

Georgia Institute of Technology  
258 4TH St  
Atlanta, GA 30332  
UNITED STATES

Administrative Contact:

Robin Greene  
Georgia Institute of Technology  
258 Fourth Street NW  
Atlanta, GA 30332-0700  
UNITED STATES  
(404) 894-6176  
hostmaster@gatech.edu

Technical Contact:

Scott Friedrich  
Georgia Institute of Technology  
258 Fourth Street NW  
Atlanta, GA 30332-0700  
UNITED STATES  
(404) 894-6720  
hostmaster@gatech.edu

Name Servers:

DNS1.GATECH.EDU	128.61.244.253
DNS2.GATECH.EDU	130.207.244.81
DNS3.GATECH.EDU	168.24.2.35

Domain record activated: 08-May-1986

Domain record last updated: 08-Feb-2016

Domain expires: 31-Jul-2016

Traceroute shows how data transmits from one to another by displaying all the computers and servers the packet travel through in the middle. With this information, attacks may be designed to focus on finding out how information travel within a group (e.g. company) and attack certain servers which are the bottlenecks in the group (servers where every computer has to pass through).

- What is the impact to end users if they query a DNS server with a poisoned cache? **[0.5 points]**

A DNS server with a poisoned cache will lead the end users to a wrong and most likely malicious server. The server can be harmful in many ways, including impersonating the intended website to steal the user's' data.

- Explain the process for performing a poisoning attack. Be sure to note why it works. **[0.5 points]**

A cache is poisoned when some user needs to issue a DNS query. The query is sent to the recursive resolver with some associated ID, and then the resolver sends that request to the root server. The recursive resolver then waits for a response from either the root DNS server, or some authoritative name server. An attacker may not know what ID is associated with the DNS query, but they can flood the resolver with several fake responses, each having a different ID. If the attacker can correctly guess the query ID before the authoritative name server can return a legitimate response, the resolver caches the attacker's falsified DNS query response. This attack works because DNS has no way to verify the legitimacy of a response. DNS must wait for the cached address to expire in order to expunge the false information.

- Now that you understand the weaknesses of the basic Domain Name System, what is one way it could be improved? (Hint: look up DNSSEC, 0x20-bit Encoding, or other techniques.) **[0.5 points]**

DNSSEC (Domain Name System Security Extensions) is a modification to the DNS protocol that allows DNS messages to be digitally signed using public-key cryptography. This mitigates the potential for a cache poisoning attack as DNS query responses can be verified as messages from legitimate authoritative name servers.

### **SSL/TLS and Traffic Analysis (6 Points)**

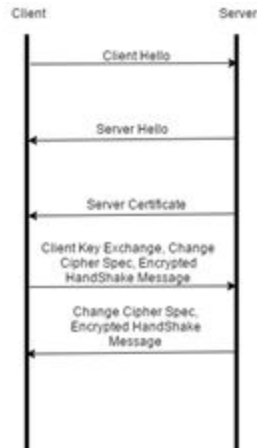
- What is different when visiting the site using HTTP versus HTTPS? **[1 point]**

HTTPS encrypts the information between the client and the server using SSL and keep the information safe, while HTTP allows the exchanges to be in plaintext. There is an additional SSL/TLS handshake procedure for HTTPS connections.

At first glance, the obvious difference is the color scheme of the packets Wireshark collects. The unsecure HTTP websites resulted in mostly green TCP packets being sent between the website and my browser. Secure HTTPS websites sent mostly purple TCP packets. The coloring rules of Wireshark indicate that the green packets are seen by Wireshark as sent using the HTTP and TCP protocol, while the purple packets only reveal that they are using the TCP protocol.

Website used: <https://www.wellsfargo.com>

- Draw a diagram showing the SSL/TLS handshake procedure you observed. Attach a screenshot of the Wireshark capture that shows the first few packets of the handshake. **[2 points]**



9	0.071462_	10.0.2.15	159.45.2.145	TCP	54 35466 - 443 [ACK] Seq=1 Ack=1 Win=29200 Len=0
10	0.071737_	10.0.2.15	159.45.2.145	TLSv1.2	246 Client Hello
11	0.072391_	159.45.2.145	10.0.2.15	TCP	60 443 - 35466 [ACK] Seq=1 Ack=193 Win=65535 Len=0
12	0.109733_	159.45.2.145	10.0.2.15	TLSv1.2	2814 Server Hello
13	0.109748_	10.0.2.15	159.45.2.145	TCP	54 35466 - 443 [ACK] Seq=193 Ack=2761 Win=34080 Len=0
14	0.110890_	159.45.2.145	10.0.2.15	TLSv1.2	1271 Certificate
15	0.110896_	10.0.2.15	159.45.2.145	TCP	54 35466 - 443 [ACK] Seq=193 Ack=3978 Win=36920 Len=0
16	0.115716_	10.0.2.15	159.45.2.145	TLSv1.2	396 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
17	0.116467_	159.45.2.145	10.0.2.15	TCP	60 443 - 35466 [ACK] Seq=3978 Ack=535 Win=65535 Len=0
18	0.155197_	159.45.2.145	10.0.2.15	TLSv1.2	129 Change Cipher Spec, Encrypted Handshake Message
19	0.155388_	10.0.2.15	159.45.2.145	TLSv1.2	763 Application Data
20	0.155990_	159.45.2.145	10.0.2.15	TCP	60 443 - 35466 [ACK] Seq=4053 Ack=1244 Win=65535 Len=0
21	0.757944_	159.45.2.145	10.0.2.15	TCP	7154 [TCP segment of a reassembled PDU]
22	0.758024_	10.0.2.15	159.45.2.145	TCP	54 35466 - 443 [ACK] Seq=1244 Ack=11153 Win=51120 Len=0
23	0.758198_	159.45.2.145	10.0.2.15	TCP	1387 [TCP segment of a reassembled PDU]
24	0.762790_	159.45.2.145	10.0.2.15	TLSv1.2	4927 Application Data
25	0.792823_	10.0.2.15	159.45.2.145	TCP	54 35466 - 443 [ACK] Seq=1244 Ack=17359 Win=63900 Len=0

- Which organization(s) issued the certificate of the web server? [1 point]

It is issued by Symantec Corporation

- What is the public RSA key of the web server? [1 point]

Modulus (2048 bits):

```

ad ab 81 2e d6 f1 d5 49 6c 38 c0 47 0b 5c 19 1d
f0 a1 75 70 3c de 9e 84 ba 62 32 35 42 9b f7 31
9f e9 02 00 e0 00 ca bf 8f d6 e8 b3 5c df 3b 3f
29 0d dc b3 9d d0 8e a9 a7 a6 06 74 af 9e 33 5b
46 0e 19 04 06 06 f7 4a be f5 4b 13 a8 4c 1d 15
e6 45 96 24 59 60 a3 e5 10 16 e9 5d 0d 63 83 27
17 8d 26 47 9a 98 76 56 ed 9f 0d 9d e0 42 ae da
0e b9 59 2c c7 9f 7c 6d 34 71 c8 ba b9 c4 d9 9a
9b cd 2c 9b f3 e8 15 30 1d 1b 68 cf cc 65 fc a9
49 40 dd c3 33 45 3a 4e 47 b0 a3 fa b8 d7 9e 77
12 7b b6 1f 39 e0 5a e2 af ff e9 60 d8 ec 21 37
22 b4 91 a2 9d ae a3 3f 8d 77 be 5a 0f 8a 84 d7
0a 0b 65 5a 60 56 f9 48 54 6b 99 8e 2d 97 71 c3
94 a5 fe 54 46 d5 52 3d 94 97 78 e2 7c f0 08 9b
4a c2 ef 81 67 ee 07 6a f1 fe e6 9a d5 70 3c f4
8b e3 1e 9d 2c ae 4f 2a fe 41 8e 60 fe d4 5e 89
  
```

Exponent (24 bits):  
65537

- If the client (your browser) cannot verify the validity of a remote server's certificate and you choose to trust it anyway, what security properties do you lose? **[1 point]**

Authentication

### **nmap Security Scanner (4 points)**

Attached output file **[0.5 points]**

- PCAP1:
  - What is the IP address of the scanner? **[0.25 points]**  
192.168.0.9
  - What is the IP address of the target? **[0.25 points]**  
192.168.0.99
  - Which ports were scanned? (Hint: how were they chosen?) **[0.25 points]**  
From 1 to 1024
  - Which port scanning technique was used? Please justify your answer and describe how the technique works. **[0.25 points]**  
TCP Xmas scan. We can observe that the (FIN, PSH, URG) are flagged in each packet sent from the source to the target.  
*"The TCP XMAS scan is used to identify listening ports on the targeted system. The scan manipulates the URG, PSH and FIN flags of the TCP header. If the port is closed on the targeted system, the target will send an RST. If the port is open, the port will ignore the packets. (sans.org)*  
*The key advantage to these scan types is that they can sneak through certain non-stateful firewalls and packet filtering routers."* (nmap.org)
  - What is the command and arguments for this scan? Your answer should be what you would type into the shell, e.g. `nmap -sT -p 443 127.0.0.1`. **[0.25 points]**  
`nmap -sX -p 1-1024 192.168.0.99/32`
- PCAP2:
  - What is the IP address of the scanner? **[0.25 points]**  
192.168.0.9  
The rest of the IPs are decoys

- What is the IP address of the target? **[0.25 points]**

192.168.0.99

- Which ports were scanned? (Hint: how were they chosen?) **[0.25 points]**

1-65535

- Which port scanning technique was used? Please justify your answer and describe how the technique works. **[0.25 points]**

TCP Xmas scan. We can observe that the (FIN, PSH, URG) are flagged in each packet sent from the source to the target.

*"The TCP XMAS scan is used to identify listening ports on the targeted system. The scan manipulates the URG, PSH and FIN flags of the TCP header. If the port is closed on the targeted system, the target will send an RST. If the port is open, the port will ignore the packets. (sans.org)*

*The key advantage to these scan types is that they can sneak through certain non-stateful firewalls and packet filtering routers." (nmap.org)*

- What is the command and arguments for this scan? Your answer should be what you would type into the shell, e.g. `nmap -sT -p 443 127.0.0.1`. **[0.25 points]**

`nmap -sX -p -D 192.168.0.1 192.168.0.1, 192.168.0.254, 192.168.0.199 192.168.0.99/32`

- Research nmap's different scan techniques. It performs the SYN scan (-sS) by default. Why? **[0.5 points]**

*SYN scan is the default and most popular scan option for good reasons. It can be performed quickly, scanning thousands of ports per second on a fast network not hampered by restrictive firewalls. It is also relatively unobtrusive and stealthy since it never completes TCP connections. SYN scan works against any compliant TCP stack rather than depending on idiosyncrasies of specific platforms as Nmap's FIN/NULL/Xmas, Maimon and idle scans do. It also allows clear, reliable differentiation between the open, closed, and filtered states.*

*This technique is often referred to as half-open scanning, because you don't open a full TCP connection. You send a SYN packet, as if you are going to open a real connection and then wait for a response. A SYN/ACK indicates the port is listening (open), while a RST (reset) is indicative of a non-listener. If no response is received after several retransmissions, the port is marked as filtered. The port is also marked filtered if an ICMP unreachable error (type 3, code 0, 1, 2, 3, 9, 10, or 13) is received. The port is also considered open if a SYN packet (without the ACK flag) is received in response. This can be due to an extremely rare TCP feature known as a simultaneous open or split handshake connection (see <http://nmap.org/misc/split-handshake.pdf>). (nmap.org)*

- Choose one other scan technique (i.e. -sF, -sA, etc.) and discuss its advantages and disadvantages. **[0.5 point]**

Another scanning technique is -sU which uses the UDP protocol. It is commonplace for security auditors to ignore the potential for UDP scans because they are slower than TCP scans. The UDP scan sends a UDP packet to every target port. Usually each packet that is sent from the scanner is empty, unless the target port is a common port or requires some protocol-specific payload. The client can recognize which ports are open, closed or filtered based on the response it receives.

The biggest challenge with UDP scans is run time. Because of the nature of the typical responses to UDP scans, waiting to confirm a no response, or receiving an ICMP port unreachable error takes much longer than TCP scans.

### iptables Firewall (4 Points)

Insert two screenshots here [0.5 points per screenshot]

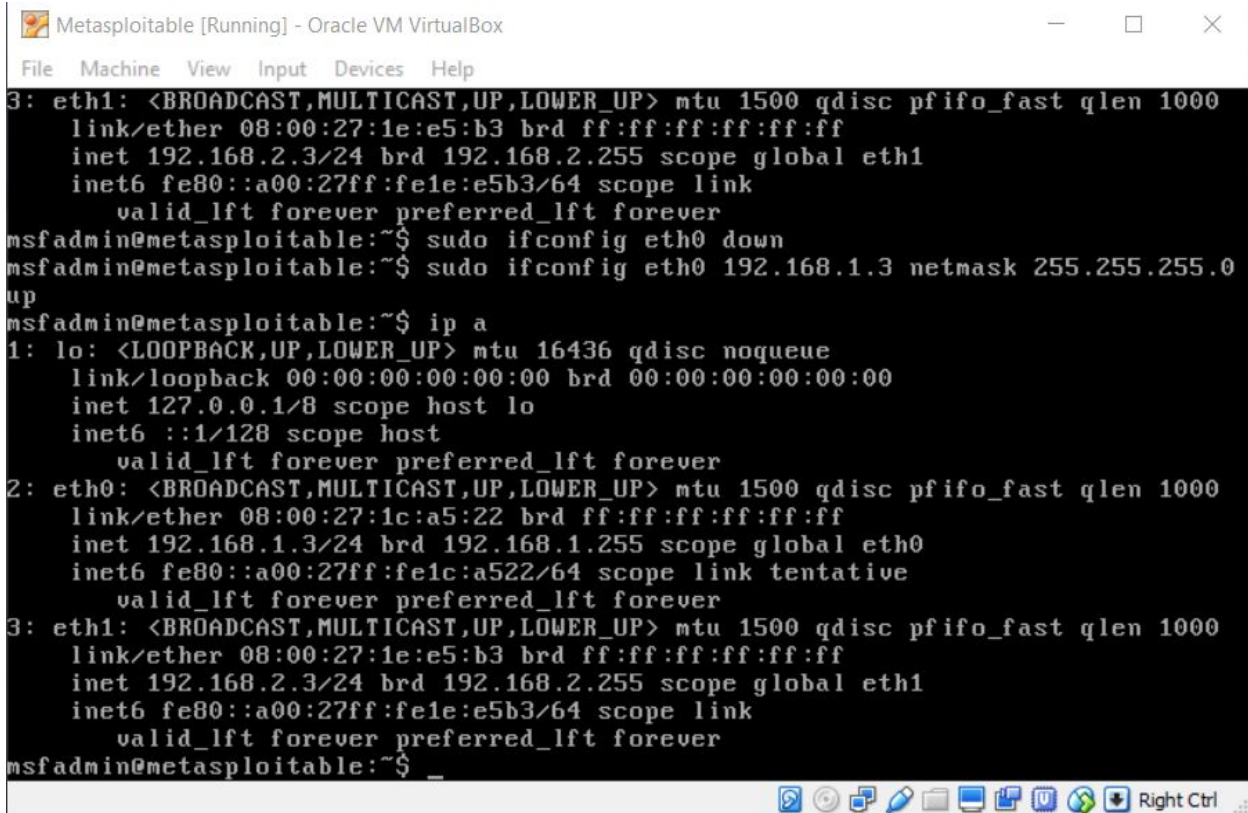
```
msfadmin@metasploitable:~$ sudo iptables -L
Chain INPUT (policy DROP)
target      prot opt source                destination          tcp dpt:ssh
ACCEPT      tcp  --  192.168.2.0/24          anywhere

Chain FORWARD (policy DROP)
target      prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target      prot opt source                destination
msfadmin@metasploitable:~$
```



```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# ^C  
root@kali:~# ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default  
    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 pfifo_fast state UP group  
    default qlen 1000  
    link/ether 08:00:27:6d:5e:27 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.1.2/24 brd 192.168.1.255 scope global eth0  
        valid_lft forever preferred_lft forever  
    inet6 fe80::a00:27ff:fe6d:5e27/64 scope link  
        valid_lft forever preferred_lft forever  
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group  
    default qlen 1000  
    link/ether 08:00:27:b0:18:53 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.2.2/24 brd 192.168.2.255 scope global eth1  
        valid_lft forever preferred_lft forever  
    inet6 fe80::a00:27ff:feb0:1853/64 scope link  
        valid_lft forever preferred_lft forever  
root@kali:~#
```



```
Metasploitable [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
  link/ether 08:00:27:1e:e5:b3 brd ff:ff:ff:ff:ff:ff
  inet 192.168.2.3/24 brd 192.168.2.255 scope global eth1
  inet6 fe80::a00:27ff:fe1e:e5b3/64 scope link
    valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$ sudo ifconfig eth0 down
msfadmin@metasploitable:~$ sudo ifconfig eth0 192.168.1.3 netmask 255.255.255.0
up
msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
  link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
  inet 127.0.0.1/8 scope host lo
  inet6 ::1/128 scope host
    valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
  link/ether 08:00:27:1c:a5:22 brd ff:ff:ff:ff:ff:ff
  inet 192.168.1.3/24 brd 192.168.1.255 scope global eth0
  inet6 fe80::a00:27ff:fe1c:a522/64 scope link tentative
    valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
  link/ether 08:00:27:1e:e5:b3 brd ff:ff:ff:ff:ff:ff
  inet 192.168.2.3/24 brd 192.168.2.255 scope global eth1
  inet6 fe80::a00:27ff:fe1e:e5b3/64 scope link
    valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$ _
```

- Is it possible to configure a firewall to block nmap scans, but allow legitimate traffic? Why or why not? **[1 point]**

No. Any ports that are open to legitimate traffic is open to scans since nmap scans can be legitimate traffic as well.

- iptables can be configured to either REJECT or DROP unwanted packets. DROP simply ignores the packet. What does REJECT do? Can you think of any circumstances where it might be better to use REJECT? **[1 point]**

REJECT rejects the connection and send back an error letting the source know that the system does not want to be connected. It sends a signal saying “you shouldn’t be here” to the source, rather than just hiding.

- Is a firewall alone sufficient to protect a network? Why or why not? **[1 point]**

A firewall has to allow some traffic to pass through so some ports will be open. Attackers can still pose as a legitimate traffic and attack on the opened ports.