Lab 3: Malware and Encryption Applications

Professor: Khalil Abuosba

Student number: 212779997

Date Due: Mar 2nd 2019

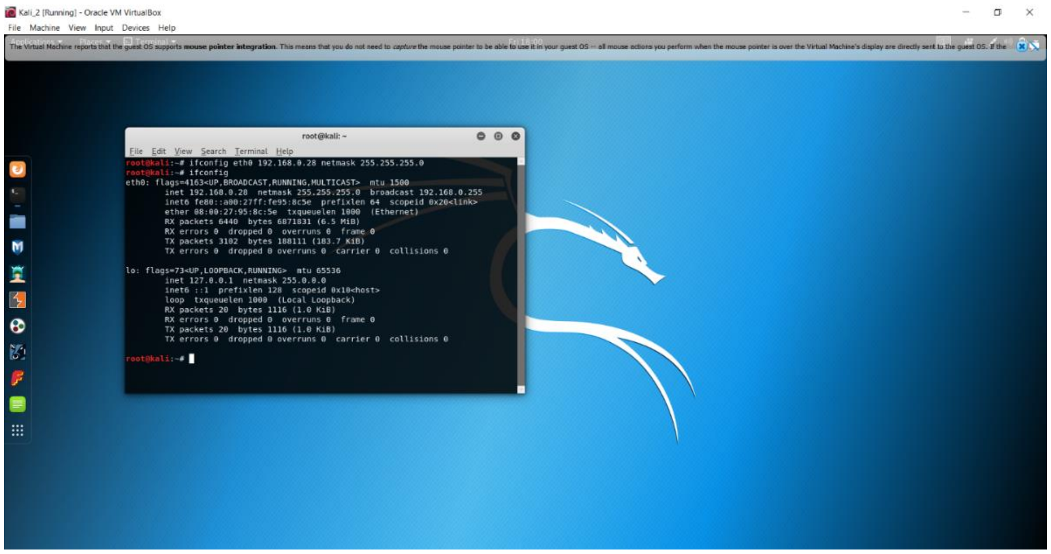
Prelab
q1-15: is done on the computer using kali linux.

Q16a: Kali-1 IPV4: 192.168.0.27

b. Kali-2 IPV4: 192.168.0.28

c. ifconfig eth0 192.168.0.28 netmask 255.255.255.0; this screenshot shows what happen when you use this ifconfig eth0 192.168.0.28 netmask 255.255.255.0.

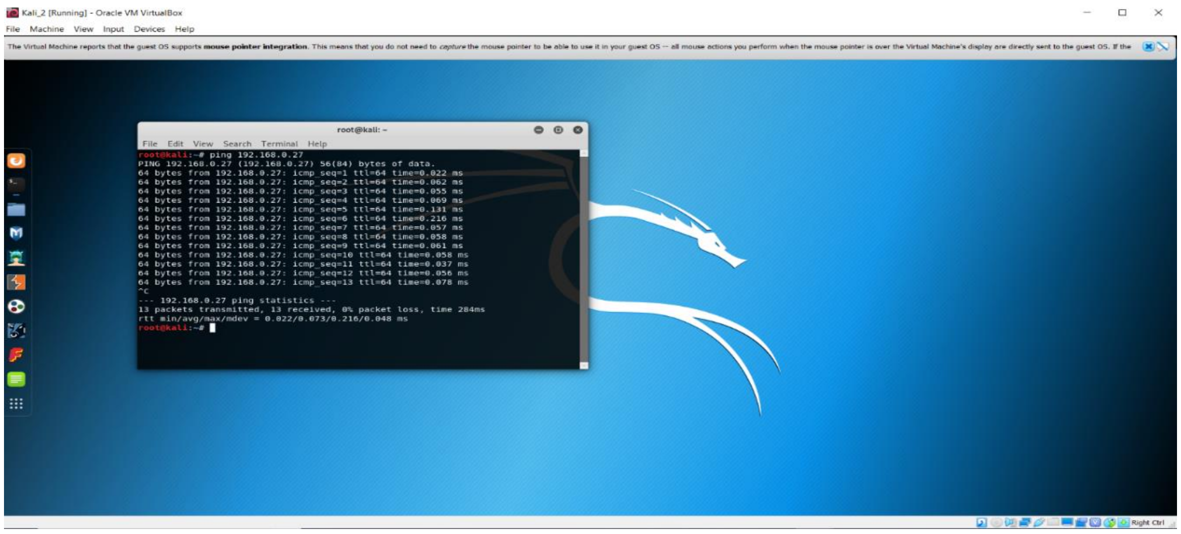
16d. after doing 16.c you find out that the ip address for kail_1,Kali_2 you get the IP address to be: 192.168.0.27 for kail_1 and 192.168.0.28 for Kali_2.


Kali_2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root@kali: ~
File Edit View Search Terminal Help

```
root@kali:~# ifconfig eth0 192.168.0.28 netmask 255.255.255.0
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.28  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::a00:27ff:fe95:8c5e  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:95:8c:5e  txqueuelen 1000  (Ethernet)
        RX packets 6440  bytes 6871831 (6.5 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3102  bytes 188111 (183.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 20  bytes 1116 (1.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 20  bytes 1116 (1.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@kali:~# 
```

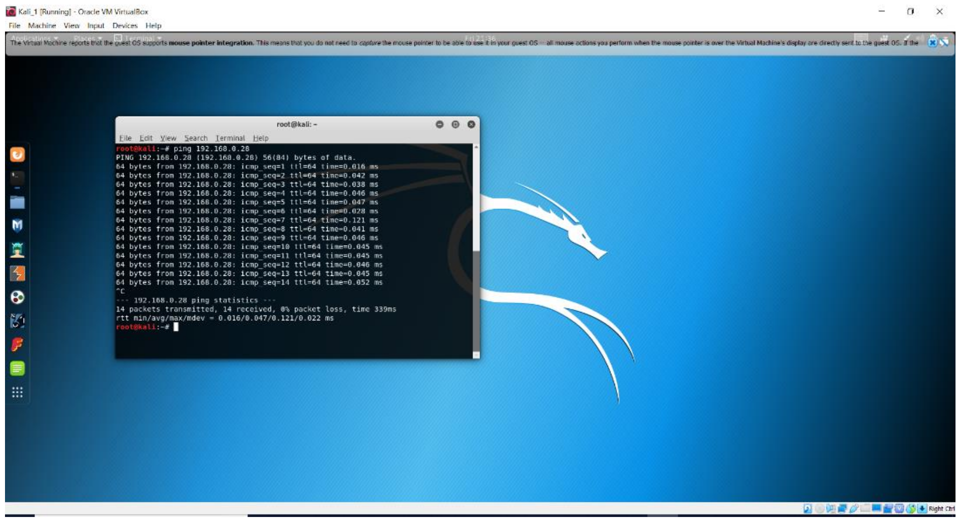
18. The loop back address for kail_1 and Kali_2 will be the same so I took one screenshot. It is below:

Q19: when I ping ip address of kali_1 in Kali_2 I get the following:


Kali_2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root@kali: ~
File Edit View Search Terminal Help

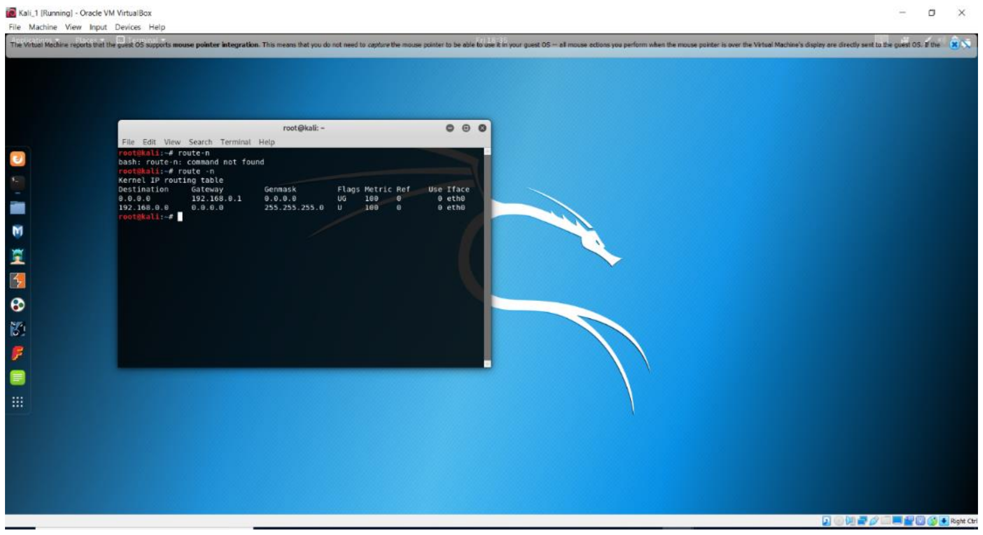
```
root@kali:~# ping 192.168.0.27
PING 192.168.0.27 (192.168.0.27) 56(84) bytes of data.
64 bytes from 192.168.0.27: icmp_seq=1 ttl=64 time=0.022 ms
64 bytes from 192.168.0.27: icmp_seq=2 ttl=64 time=0.062 ms
64 bytes from 192.168.0.27: icmp_seq=3 ttl=64 time=0.055 ms
64 bytes from 192.168.0.27: icmp_seq=4 ttl=64 time=0.069 ms
64 bytes from 192.168.0.27: icmp_seq=5 ttl=64 time=0.131 ms
64 bytes from 192.168.0.27: icmp_seq=6 ttl=64 time=0.216 ms
64 bytes from 192.168.0.27: icmp_seq=7 ttl=64 time=0.057 ms
64 bytes from 192.168.0.27: icmp_seq=8 ttl=64 time=0.058 ms
64 bytes from 192.168.0.27: icmp_seq=9 ttl=64 time=0.061 ms
64 bytes from 192.168.0.27: icmp_seq=10 ttl=64 time=0.058 ms
64 bytes from 192.168.0.27: icmp_seq=11 ttl=64 time=0.037 ms
64 bytes from 192.168.0.27: icmp_seq=12 ttl=64 time=0.056 ms
64 bytes from 192.168.0.27: icmp_seq=13 ttl=64 time=0.078 ms
^C
--- 192.168.0.27 ping statistics ---
13 packets transmitted, 13 received, 0% packet loss, time 284ms
rtt min/avg/max/mdev = 0.022/0.073/0.216/0.048 ms
root@kali:~# 
```

When I ping Kali_2 IP address into Kali_1 I get:

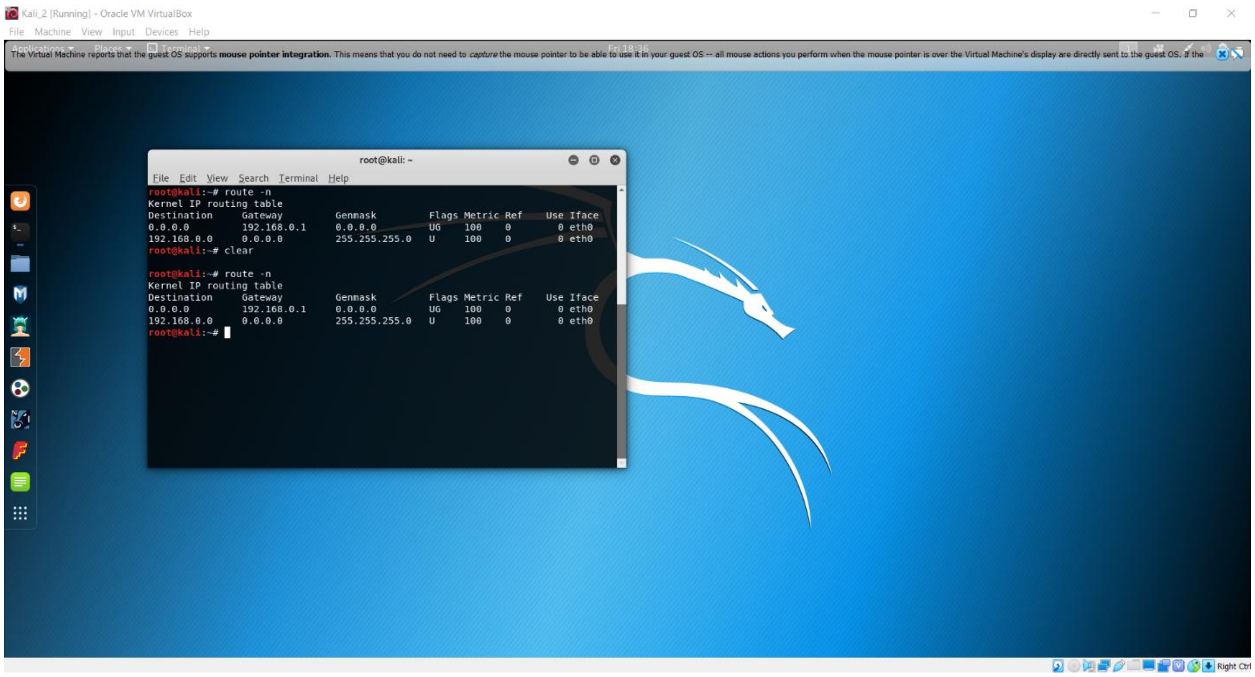


```
root@kali:~# ping 192.168.0.28
PING 192.168.0.28 (192.168.0.28) 56(84) bytes of data.
64 bytes from 192.168.0.28: icmp_seq=1 ttl=64 time=0.016 ms
64 bytes from 192.168.0.28: icmp_seq=2 ttl=64 time=0.042 ms
64 bytes from 192.168.0.28: icmp_seq=3 ttl=64 time=0.038 ms
64 bytes from 192.168.0.28: icmp_seq=4 ttl=64 time=0.046 ms
64 bytes from 192.168.0.28: icmp_seq=5 ttl=64 time=0.047 ms
64 bytes from 192.168.0.28: icmp_seq=6 ttl=64 time=0.028 ms
64 bytes from 192.168.0.28: icmp_seq=7 ttl=64 time=0.121 ms
64 bytes from 192.168.0.28: icmp_seq=8 ttl=64 time=0.041 ms
64 bytes from 192.168.0.28: icmp_seq=9 ttl=64 time=0.046 ms
64 bytes from 192.168.0.28: icmp_seq=10 ttl=64 time=0.045 ms
64 bytes from 192.168.0.28: icmp_seq=11 ttl=64 time=0.045 ms
64 bytes from 192.168.0.28: icmp_seq=12 ttl=64 time=0.046 ms
64 bytes from 192.168.0.28: icmp_seq=13 ttl=64 time=0.045 ms
64 bytes from 192.168.0.28: icmp_seq=14 ttl=64 time=0.052 ms
^C
--- 192.168.0.28 ping statistics ---
14 packets transmitted, 14 received, 0% packet loss, time 339ms
rtt min/avg/max/mdev = 0.016/0.047/0.121/0.022 ms
root@kali:~#
```

Q20. IP routing table for both servers is shown below:



```
root@kali:~# route-n
bash: route-n: command not found
root@kali:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.0.1     0.0.0.0         UG    100    0        0 eth0
192.168.0.0     0.0.0.0         255.255.255.0   U     100    0        0 eth0
root@kali:~#
```

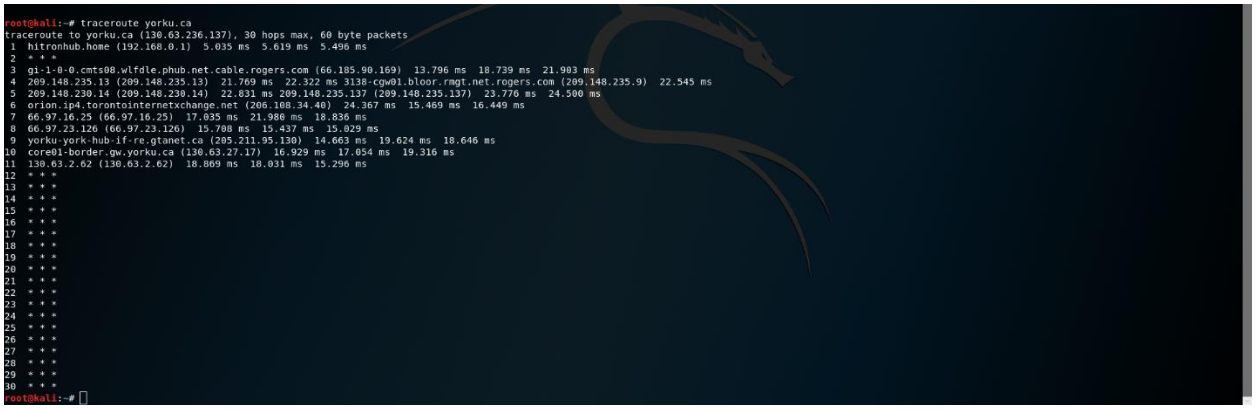


```
root@kali:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.0.1     0.0.0.0         UG    100    0        0 eth0
192.168.0.0     0.0.0.0         255.255.255.0   U     100    0        0 eth0
root@kali:~# clear

root@kali:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.0.1     0.0.0.0         UG    100    0        0 eth0
192.168.0.0     0.0.0.0         255.255.255.0   U     100    0        0 eth0
root@kali:~#
```

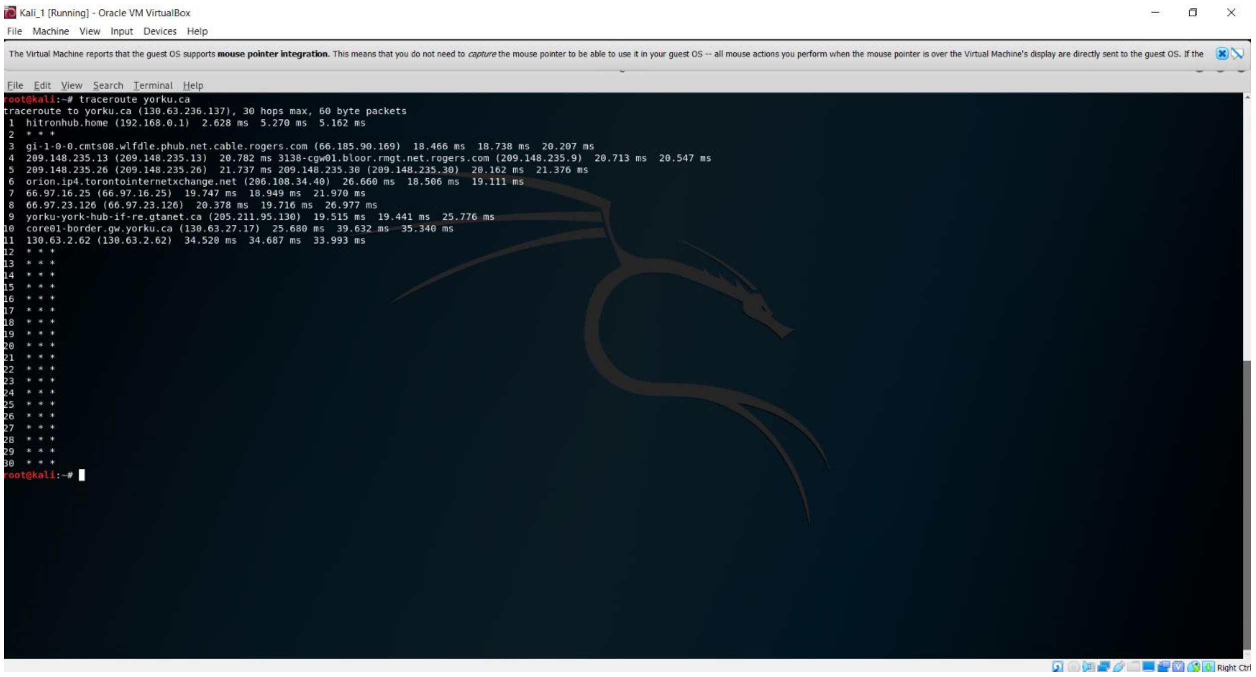
Q21. The default IPv4 address for the default gateway is: 192.168.0.1 for both.

Q22.

Q24c.



```
root@kali:~# traceroute yorku.ca
traceroute to yorku.ca (130.63.236.137), 30 hops max, 60 byte packets
 1  hitronhub.home (192.168.0.1)  5.035 ms  5.619 ms  5.496 ms
 2  * * *
 3  gi-1-0-0.cmts08.wlfdle.phub.net.cable.rogers.com (66.185.90.169)  13.796 ms  18.739 ms  21.903 ms
 4  209.148.235.13 (209.148.235.13)  21.769 ms  22.322 ms 3138-cgw01.bloor.rmgt.net.rogers.com (209.148.235.9)  22.545 ms
 5  209.148.230.14 (209.148.230.14)  22.831 ms 209.148.235.137 (209.148.235.137)  23.776 ms  24.500 ms
 6  orion.ip4.torontointernetxchange.net (206.108.34.40)  24.367 ms  15.469 ms  16.449 ms
 7  66.97.16.25 (66.97.16.25)  17.035 ms  21.980 ms  18.836 ms
 8  66.97.23.126 (66.97.23.126)  15.708 ms  15.437 ms  15.029 ms
 9  yorku-york-hub-if-re.gtanet.ca (205.211.95.130)  14.663 ms  19.624 ms  18.646 ms
10  core01-border.gw.yorku.ca (130.63.27.17)  16.929 ms  17.054 ms  19.316 ms
11  130.63.2.62 (130.63.2.62)  18.869 ms  18.031 ms  15.296 ms
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
root@kali:~#
```


Kali_1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

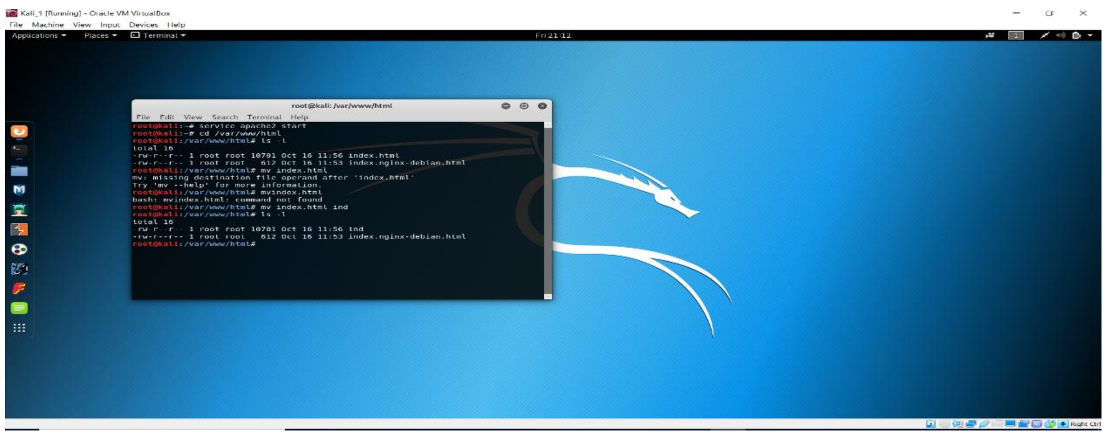
The Virtual Machine reports that the guest OS supports **mouse pointer integration.** This means that you do not need to *capture* the mouse pointer to be able to use it in your guest OS -- all mouse actions you perform when the mouse pointer is over the Virtual Machine's display are directly sent to the guest OS. If the

```
root@kali:~# traceroute yorku.ca
traceroute to yorku.ca (130.63.236.137), 30 hops max, 60 byte packets
 1  hitronhub.home (192.168.0.1)  2.628 ms  5.270 ms  5.162 ms
 2  * * *
 3  gi-1-0-0.cmts08.wlfdle.phub.net.cable.rogers.com (66.185.90.169)  18.466 ms  18.738 ms  20.207 ms
 4  209.148.235.13 (209.148.235.13)  20.782 ms 3138-cgw01.bloor.rmgt.net.rogers.com (209.148.235.9)  20.713 ms  20.547 ms
 5  209.148.235.26 (209.148.235.26)  21.737 ms 209.148.235.30 (209.148.235.30)  20.162 ms  21.376 ms
 6  orion.ip4.torontointernetxchange.net (206.108.34.40)  26.660 ms  18.506 ms  19.111 ms
 7  66.97.16.25 (66.97.16.25)  19.747 ms  18.949 ms  21.970 ms
 8  66.97.23.126 (66.97.23.126)  20.378 ms  19.716 ms  26.977 ms
 9  yorku-york-hub-if-re.gtanet.ca (205.211.95.130)  19.515 ms  19.441 ms  25.776 ms
10  core01-border.gw.yorku.ca (130.63.27.17)  25.680 ms  39.632 ms  35.340 ms
11  130.63.2.62 (130.63.2.62)  34.520 ms  34.687 ms  33.993 ms
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
root@kali:~# 
```

Lab 3(p1): Apache Basic Authentication:

1.1-1.2: is done on kali-linux by following the lab instructions.

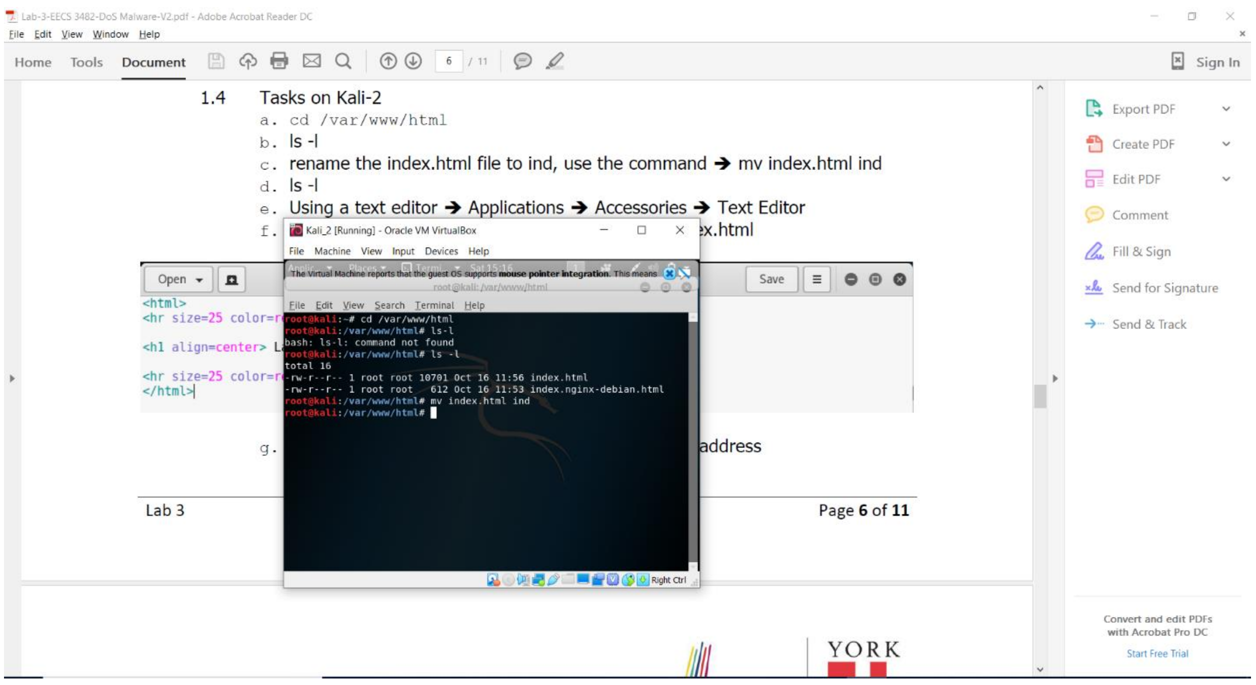
1.3a-d is shown below:


Kali_1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal
Fri 21:12
root@kali: /var/www/html

```
root@kali:~# service apache2 start
root@kali:~# cd /var/www/html
root@kali:/var/www/html# ls -l
total 16
-rw-r--r-- 1 root root 10701 Oct 16 11:56 index.html
-rw-r--r-- 1 root root   612 Oct 16 11:53 index.nginx-debian.html
root@kali:/var/www/html# mv index.html
mv: missing destination file operand after 'index.html'
Try 'mv --help' for more information.
root@kali:/var/www/html# mvindex.html
bash: mvindex.html: command not found
root@kali:/var/www/html# mv index.html ind
root@kali:/var/www/html# ls -l
total 16
-rw-r--r-- 1 root root 10701 Oct 16 11:56 ind
-rw-r--r-- 1 root root   612 Oct 16 11:53 index.nginx-debian.html
root@kali:/var/www/html# 
```

1.3g-h was done one kali_1 and Kali_2. (computer caused a crash and I wasn't able to get screenshots)

1.4a-d is shown below:



1.4 Tasks on Kali-2

a. cd /var/www/html

b. ls -l

c. rename the index.html file to ind, use the command ➔ mv index.html ind

d. ls -l

e. Using a text editor ➔ Applications ➔ Accessories ➔ Text Editor

f. …x.html

```
<html>
<hr size=25 color=r
<h1 align=center> L

<hr size=25 color=r
</html>
```

```
root@kali:~# cd /var/www/html
root@kali:/var/www/html# ls-l
bash: ls-l: command not found
root@kali:/var/www/html# ls -l
total 16
-rw-r--r-- 1 root root 10701 Oct 16 11:56 index.html
-rw-r--r-- 1 root root   612 Oct 16 11:53 index.nginx-debian.html
root@kali:/var/www/html# mv index.html ind
root@kali:/var/www/html# 
```

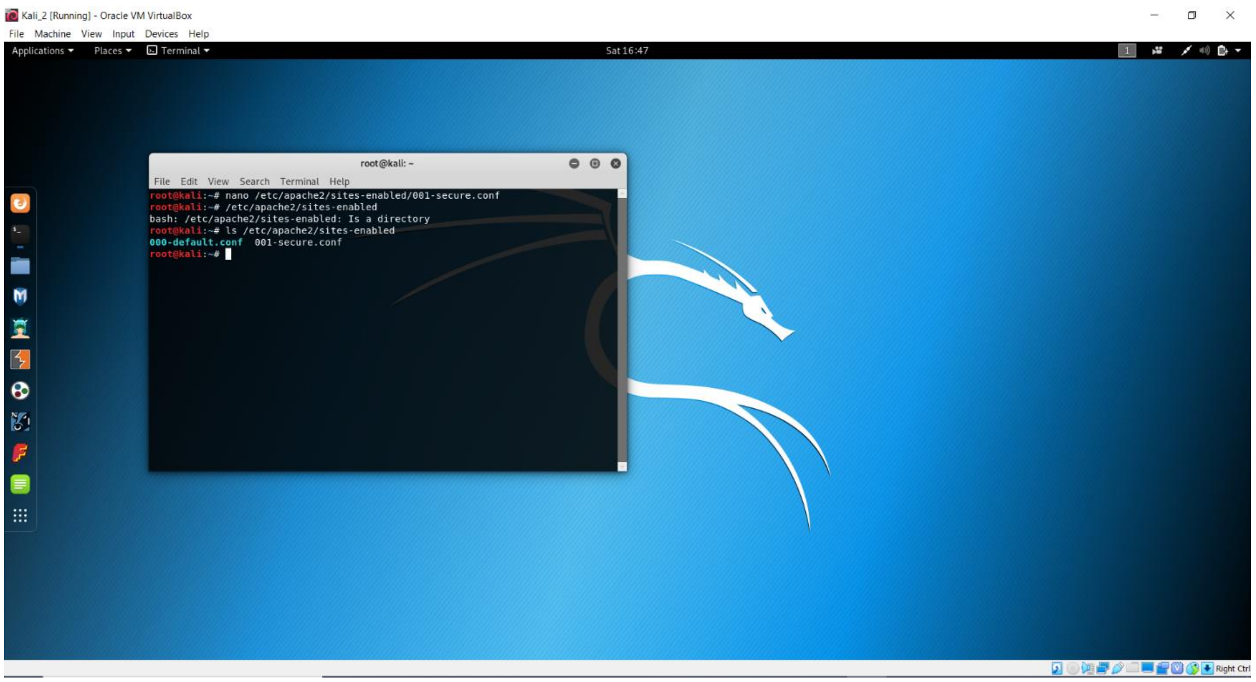
g. …address

Lab 3 Page 6 of 11

1.4g-h was done one kali_1 and Kali_2. (computer caused a crash and I wasn’t able to get screenshots)

1.5-1.6: was done on the computer using Kali linux.

1.7 Screen shot showing the files that are needed using the command in the screenshot:



```
root@kali:~# nano /etc/apache2/sites-enabled/001-secure.conf
root@kali:~# /etc/apache2/sites-enabled
bash: /etc/apache2/sites-enabled: Is a directory
root@kali:~# ls /etc/apache2/sites-enabled
000-default.conf  001-secure.conf
root@kali:~# 
```

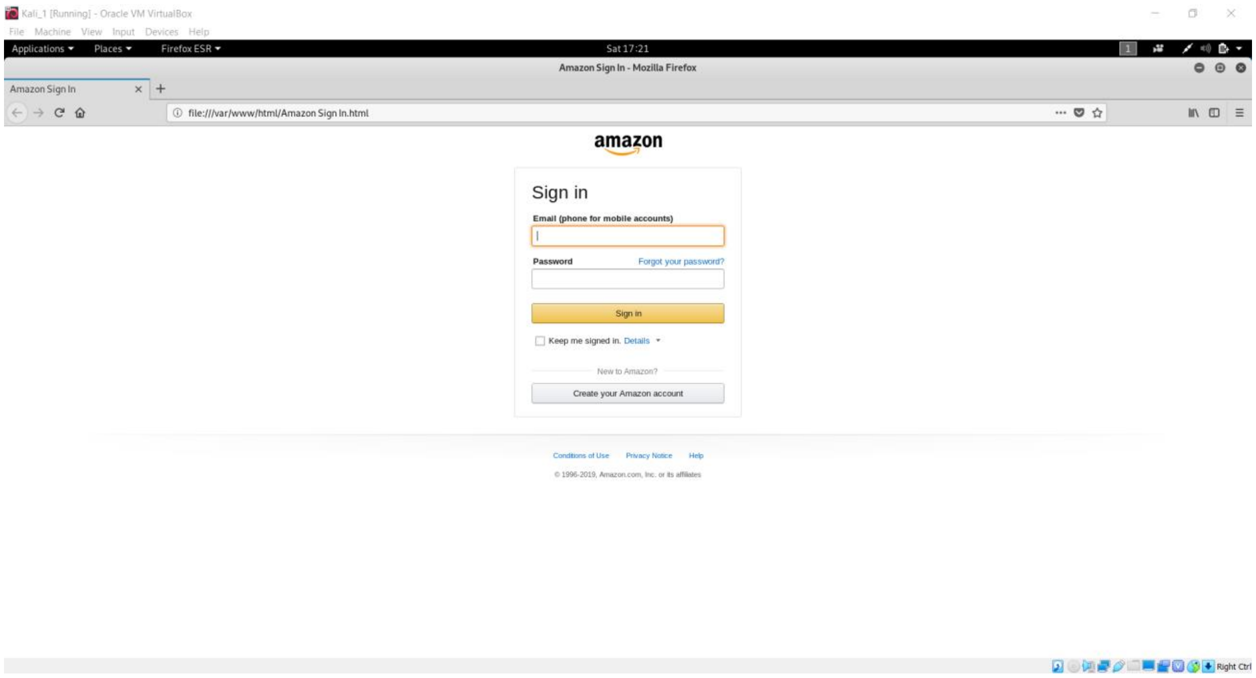
1.8-1.9: all done on kail linux;

1.10: screen shot shows the contents of .htaccess file

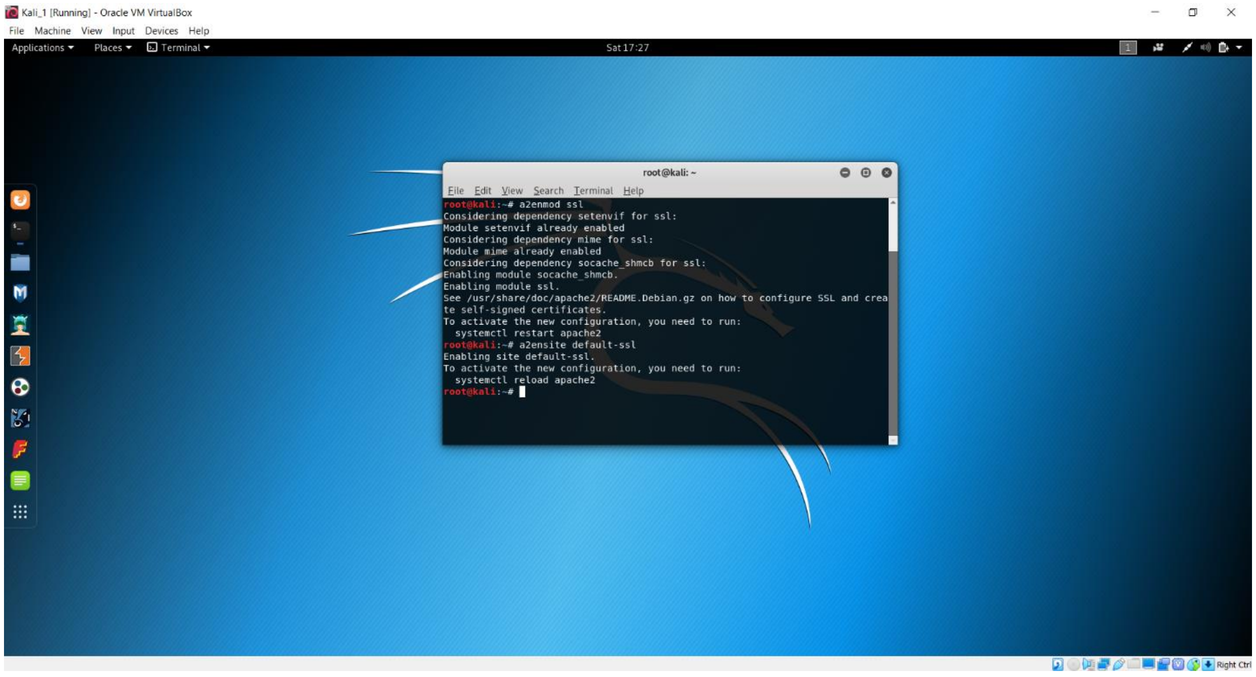
1.11-1.13: all done on kali linux

Part 2- HTTPS:

2.1: Screen shot below shows what was needed to be done:

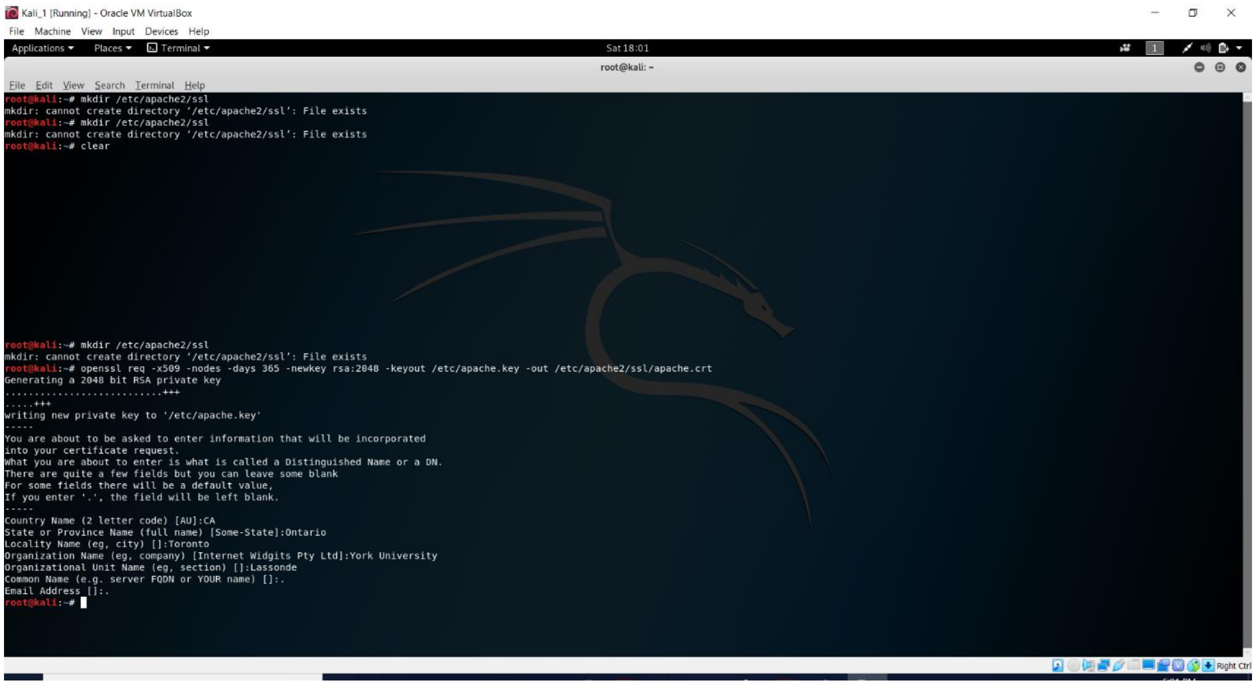

Kali_1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Firefox ESR
Sat 17:21
Amazon Sign In - Mozilla Firefox
Amazon Sign In
file:///var/www/html/Amazon Sign In.html
amazon
Sign in
Email (phone for mobile accounts)
Password
Forgot your password?
Sign in
Keep me signed in. Details
New to Amazon?
Create your Amazon account
Conditions of Use
Privacy Notice
Help
© 1996-2019, Amazon.com, Inc. or its affiliates
Right Ctrl

2.2.1: Screen shot below shows what was needed to be done:


Kali_1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal
Sat 17:27
root@kali: ~
File Edit View Search Terminal Help
Right Ctrl

```
root@kali:~# a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and crea
te self-signed certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@kali:~# a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
  systemctl reload apache2
root@kali:~#
```

2.2.2 Screen shot below shows what was needed to be done:

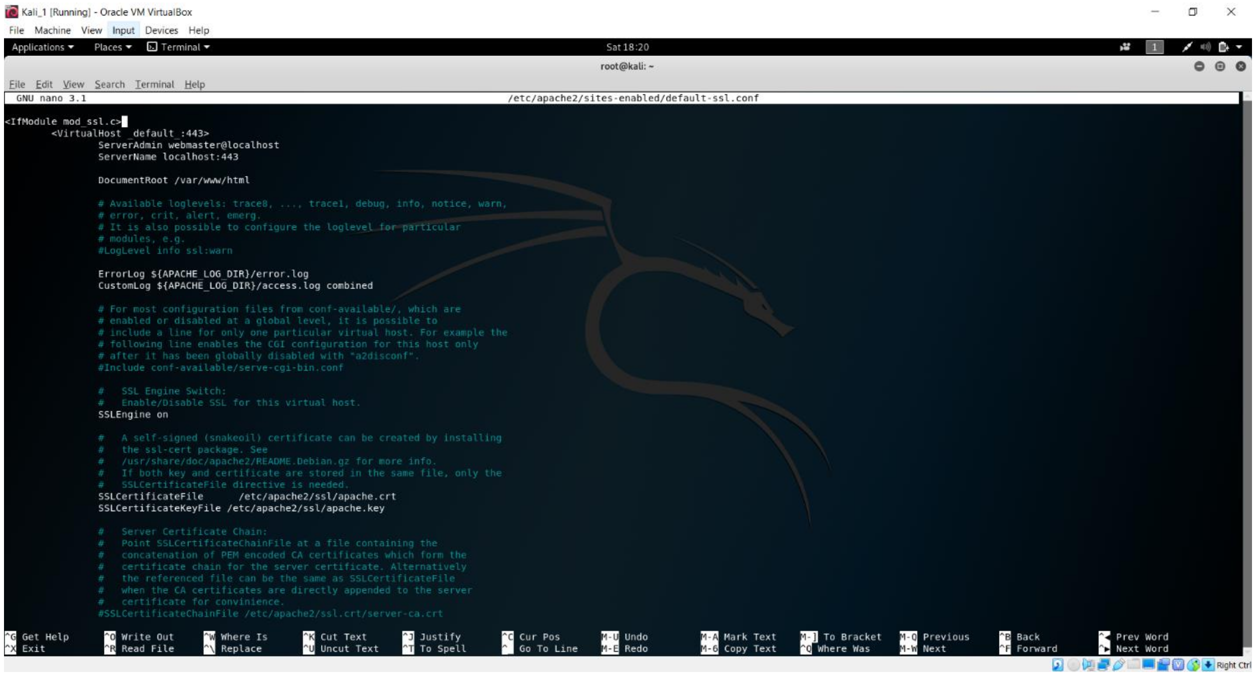


```
root@kali:~# mkdir /etc/apache2/ssl
mkdir: cannot create directory '/etc/apache2/ssl': File exists
root@kali:~# mkdir /etc/apache2/ssl
mkdir: cannot create directory '/etc/apache2/ssl': File exists
root@kali:~# clear
```

```
root@kali:~# mkdir /etc/apache2/ssl
mkdir: cannot create directory '/etc/apache2/ssl': File exists
root@kali:~# openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache.key -out /etc/apache2/ssl/apache.crt
Generating a 2048 bit RSA private key
...........................+++
.....+++
writing new private key to '/etc/apache.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:CA
State or Province Name (full name) [Some-State]:Ontario
Locality Name (eg, city) []:Toronto
Organization Name (eg, company) [Internet Widgits Pty Ltd]:York University
Organizational Unit Name (eg, section) []:Lassonde
Common Name (e.g. server FQDN or YOUR name) []:.
Email Address []:.
root@kali:~#
```

2.2.3: was done on kali linux.

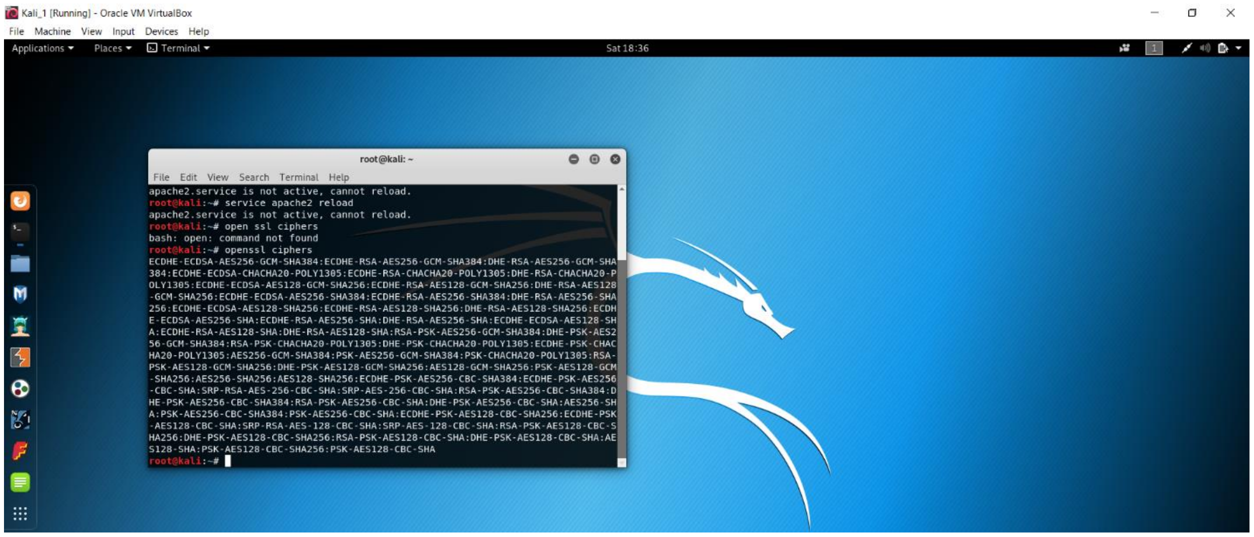
2.2.4 Screen shot below shows what was needed to be done:



```
  GNU nano 3.1                    /etc/apache2/sites-enabled/default-ssl.conf

<IfModule mod_ssl.c>
        <VirtualHost _default_:443>
                ServerAdmin webmaster@localhost
                ServerName localhost:443

                DocumentRoot /var/www/html

                # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
                # error, crit, alert, emerg.
                # It is also possible to configure the loglevel for particular
                # modules, e.g.
                #LogLevel info ssl:warn

                ErrorLog ${APACHE_LOG_DIR}/error.log
                CustomLog ${APACHE_LOG_DIR}/access.log combined

                # For most configuration files from conf-available/, which are
                # enabled or disabled at a global level, it is possible to
                # include a line for only one particular virtual host. For example the
                # following line enables the CGI configuration for this host only
                # after it has been globally disabled with "a2disconf".
                #Include conf-available/serve-cgi-bin.conf

                #   SSL Engine Switch:
                #   Enable/Disable SSL for this virtual host.
                SSLEngine on

                #   A self-signed (snakeoil) certificate can be created by installing
                #   the ssl-cert package. See
                #   /usr/share/doc/apache2/README.Debian.gz for more info.
                #   If both key and certificate are stored in the same file, only the
                #   SSLCertificateFile directive is needed.
                SSLCertificateFile      /etc/apache2/ssl/apache.crt
                SSLCertificateKeyFile /etc/apache2/ssl/apache.key

                #   Server Certificate Chain:
                #   Point SSLCertificateChainFile at a file containing the
                #   concatenation of PEM encoded CA certificates which form the
                #   certificate chain for the server certificate. Alternatively
                #   the referenced file can be the same as SSLCertificateFile
                #   when the CA certificates are directly appended to the server
                #   certificate for convinience.
                #SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.crt
```

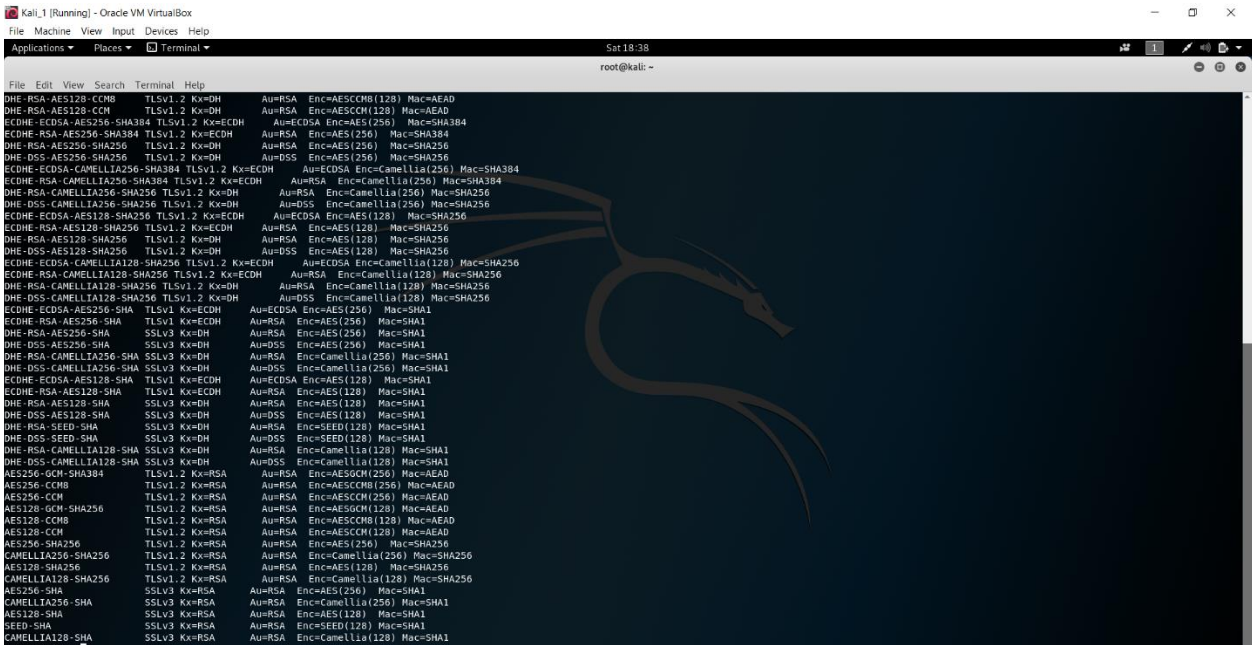
2.2.5: done on kali linux.

2.2.6a) Screen shot below shows what was needed to be done:


Kali_1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal
Sat 18:36
root@kali: ~
File Edit View Search Terminal Help

```
apache2.service is not active, cannot reload.
root@kali:~# service apache2 reload
apache2.service is not active, cannot reload.
root@kali:~# open ssl ciphers
bash: open: command not found
root@kali:~# openssl ciphers
ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-GCM-SHA
384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-CHACHA20-P
OLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES128
-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:DHE-RSA-AES256-SHA
256:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256:DHE-RSA-AES128-SHA256:ECDH
E-ECDSA-AES256-SHA:ECDHE-RSA-AES256-SHA:DHE-RSA-AES256-SHA:ECDHE-ECDSA-AES128-SH
A:ECDHE-RSA-AES128-SHA:DHE-RSA-AES128-SHA:RSA-PSK-AES256-GCM-SHA384:DHE-PSK-AES2
56-GCM-SHA384:RSA-PSK-CHACHA20-POLY1305:DHE-PSK-CHACHA20-POLY1305:ECDHE-PSK-CHAC
HA20-POLY1305:AES256-GCM-SHA384:PSK-AES256-GCM-SHA384:PSK-CHACHA20-POLY1305:RSA-
PSK-AES128-GCM-SHA256:DHE-PSK-AES128-GCM-SHA256:AES128-GCM-SHA256:PSK-AES128-GCM
-SHA256:AES256-SHA256:AES128-SHA256:ECDHE-PSK-AES256-CBC-SHA384:ECDHE-PSK-AES256
-CBC-SHA:SRP-RSA-AES-256-CBC-SHA:SRP-AES-256-CBC-SHA:RSA-PSK-AES256-CBC-SHA384:D
HE-PSK-AES256-CBC-SHA384:RSA-PSK-AES256-CBC-SHA:DHE-PSK-AES256-CBC-SHA:AES256-SH
A:PSK-AES256-CBC-SHA384:PSK-AES256-CBC-SHA:ECDHE-PSK-AES128-CBC-SHA256:ECDHE-PSK
-AES128-CBC-SHA:SRP-RSA-AES-128-CBC-SHA:SRP-AES-128-CBC-SHA:RSA-PSK-AES128-CBC-S
HA256:DHE-PSK-AES128-CBC-SHA256:RSA-PSK-AES128-CBC-SHA:DHE-PSK-AES128-CBC-SHA:AE
S128-SHA:PSK-AES128-CBC-SHA256:PSK-AES128-CBC-SHA
root@kali:~#
```

2.2.6b) Screen shot below shows what was needed to be done:

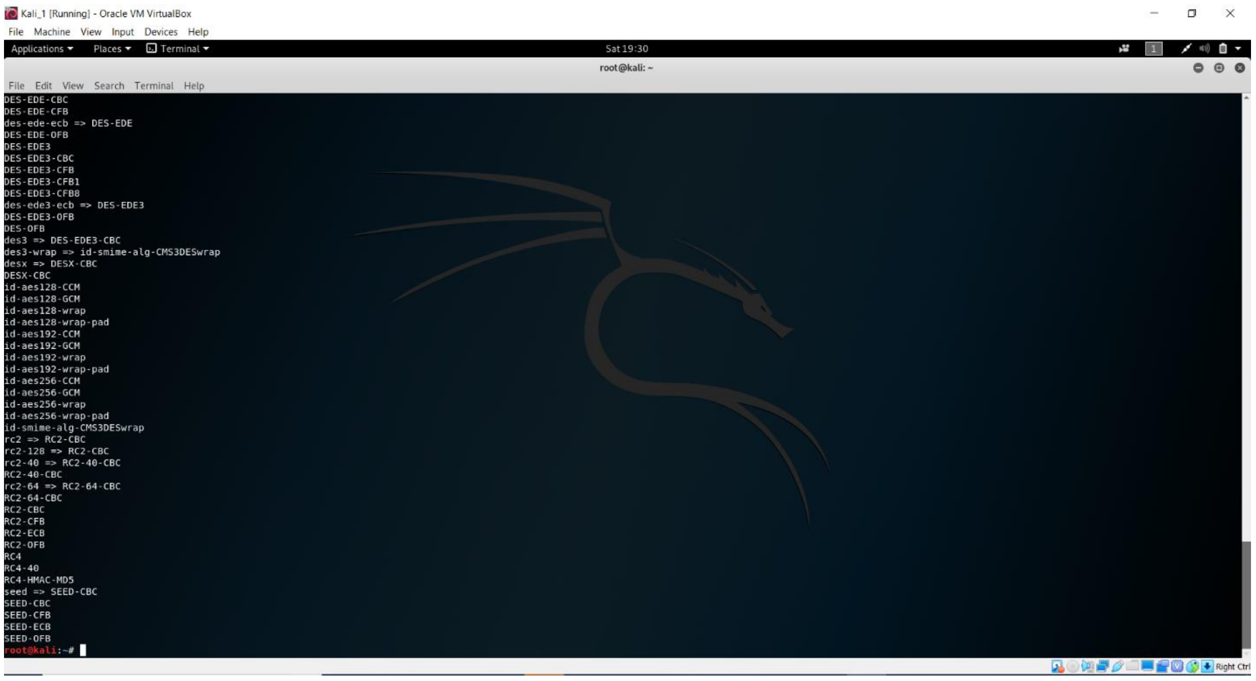

Kali_1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal
Sat 18:38
root@kali: ~
File Edit View Search Terminal Help

```
DHE-RSA-AES128-CCM8     TLSv1.2 Kx=DH       Au=RSA  Enc=AESCCM8(128) Mac=AEAD
DHE-RSA-AES128-CCM      TLSv1.2 Kx=DH       Au=RSA  Enc=AESCCM(128) Mac=AEAD
ECDHE-ECDSA-AES256-SHA384 TLSv1.2 Kx=ECDH     Au=ECDSA Enc=AES(256)  Mac=SHA384
ECDHE-RSA-AES256-SHA384 TLSv1.2 Kx=ECDH     Au=RSA  Enc=AES(256)  Mac=SHA384
DHE-RSA-AES256-SHA256   TLSv1.2 Kx=DH       Au=RSA  Enc=AES(256)  Mac=SHA256
DHE-DSS-AES256-SHA256   TLSv1.2 Kx=DH       Au=DSS  Enc=AES(256)  Mac=SHA256
ECDHE-ECDSA-CAMELLIA256-SHA384 TLSv1.2 Kx=ECDH     Au=ECDSA Enc=Camellia(256) Mac=SHA384
ECDHE-RSA-CAMELLIA256-SHA384 TLSv1.2 Kx=ECDH     Au=RSA  Enc=Camellia(256) Mac=SHA384
DHE-RSA-CAMELLIA256-SHA256 TLSv1.2 Kx=DH       Au=RSA  Enc=Camellia(256) Mac=SHA256
DHE-DSS-CAMELLIA256-SHA256 TLSv1.2 Kx=DH       Au=DSS  Enc=Camellia(256) Mac=SHA256
ECDHE-ECDSA-AES128-SHA256 TLSv1.2 Kx=ECDH     Au=ECDSA Enc=AES(128)  Mac=SHA256
ECDHE-RSA-AES128-SHA256 TLSv1.2 Kx=ECDH     Au=RSA  Enc=AES(128)  Mac=SHA256
DHE-RSA-AES128-SHA256   TLSv1.2 Kx=DH       Au=RSA  Enc=AES(128)  Mac=SHA256
DHE-DSS-AES128-SHA256   TLSv1.2 Kx=DH       Au=DSS  Enc=AES(128)  Mac=SHA256
ECDHE-ECDSA-CAMELLIA128-SHA256 TLSv1.2 Kx=ECDH     Au=ECDSA Enc=Camellia(128) Mac=SHA256
ECDHE-RSA-CAMELLIA128-SHA256 TLSv1.2 Kx=ECDH     Au=RSA  Enc=Camellia(128) Mac=SHA256
DHE-RSA-CAMELLIA128-SHA256 TLSv1.2 Kx=DH       Au=RSA  Enc=Camellia(128) Mac=SHA256
DHE-DSS-CAMELLIA128-SHA256 TLSv1.2 Kx=DH       Au=DSS  Enc=Camellia(128) Mac=SHA256
ECDHE-ECDSA-AES256-SHA  TLSv1 Kx=ECDH     Au=ECDSA Enc=AES(256)  Mac=SHA1
ECDHE-RSA-AES256-SHA    TLSv1 Kx=ECDH     Au=RSA  Enc=AES(256)  Mac=SHA1
DHE-RSA-AES256-SHA      SSLv3 Kx=DH       Au=RSA  Enc=AES(256)  Mac=SHA1
DHE-DSS-AES256-SHA      SSLv3 Kx=DH       Au=DSS  Enc=AES(256)  Mac=SHA1
DHE-RSA-CAMELLIA256-SHA SSLv3 Kx=DH       Au=RSA  Enc=Camellia(256) Mac=SHA1
DHE-DSS-CAMELLIA256-SHA SSLv3 Kx=DH       Au=DSS  Enc=Camellia(256) Mac=SHA1
ECDHE-ECDSA-AES128-SHA  TLSv1 Kx=ECDH     Au=ECDSA Enc=AES(128)  Mac=SHA1
ECDHE-RSA-AES128-SHA    TLSv1 Kx=ECDH     Au=RSA  Enc=AES(128)  Mac=SHA1
DHE-RSA-AES128-SHA      SSLv3 Kx=DH       Au=RSA  Enc=AES(128)  Mac=SHA1
DHE-DSS-AES128-SHA      SSLv3 Kx=DH       Au=DSS  Enc=AES(128)  Mac=SHA1
DHE-RSA-SEED-SHA        SSLv3 Kx=DH       Au=RSA  Enc=SEED(128) Mac=SHA1
DHE-DSS-SEED-SHA        SSLv3 Kx=DH       Au=DSS  Enc=SEED(128) Mac=SHA1
DHE-RSA-CAMELLIA128-SHA SSLv3 Kx=DH       Au=RSA  Enc=Camellia(128) Mac=SHA1
DHE-DSS-CAMELLIA128-SHA SSLv3 Kx=DH       Au=DSS  Enc=Camellia(128) Mac=SHA1
AES256-GCM-SHA384       TLSv1.2 Kx=RSA      Au=RSA  Enc=AESGCM(256) Mac=AEAD
AES256-CCM8             TLSv1.2 Kx=RSA      Au=RSA  Enc=AESCCM8(256) Mac=AEAD
AES256-CCM              TLSv1.2 Kx=RSA      Au=RSA  Enc=AESCCM(256) Mac=AEAD
AES128-GCM-SHA256       TLSv1.2 Kx=RSA      Au=RSA  Enc=AESGCM(128) Mac=AEAD
AES128-CCM8             TLSv1.2 Kx=RSA      Au=RSA  Enc=AESCCM8(128) Mac=AEAD
AES128-CCM              TLSv1.2 Kx=RSA      Au=RSA  Enc=AESCCM(128) Mac=AEAD
AES256-SHA256           TLSv1.2 Kx=RSA      Au=RSA  Enc=AES(256)  Mac=SHA256
CAMELLIA256-SHA256      TLSv1.2 Kx=RSA      Au=RSA  Enc=Camellia(256) Mac=SHA256
AES128-SHA256           TLSv1.2 Kx=RSA      Au=RSA  Enc=AES(128)  Mac=SHA256
CAMELLIA128-SHA256      TLSv1.2 Kx=RSA      Au=RSA  Enc=Camellia(128) Mac=SHA256
AES256-SHA              SSLv3 Kx=RSA      Au=RSA  Enc=AES(256)  Mac=SHA1
CAMELLIA256-SHA         SSLv3 Kx=RSA      Au=RSA  Enc=Camellia(256) Mac=SHA1
AES128-SHA              SSLv3 Kx=RSA      Au=RSA  Enc=AES(128)  Mac=SHA1
SEED-SHA                SSLv3 Kx=RSA      Au=RSA  Enc=SEED(128) Mac=SHA1
CAMELLIA128-SHA         SSLv3 Kx=RSA      Au=RSA  Enc=Camellia(128) Mac=SHA1
```

2.2.6D) Done on kali linux

2.2.6e1.7 Screen shot below shows what was needed to be done:

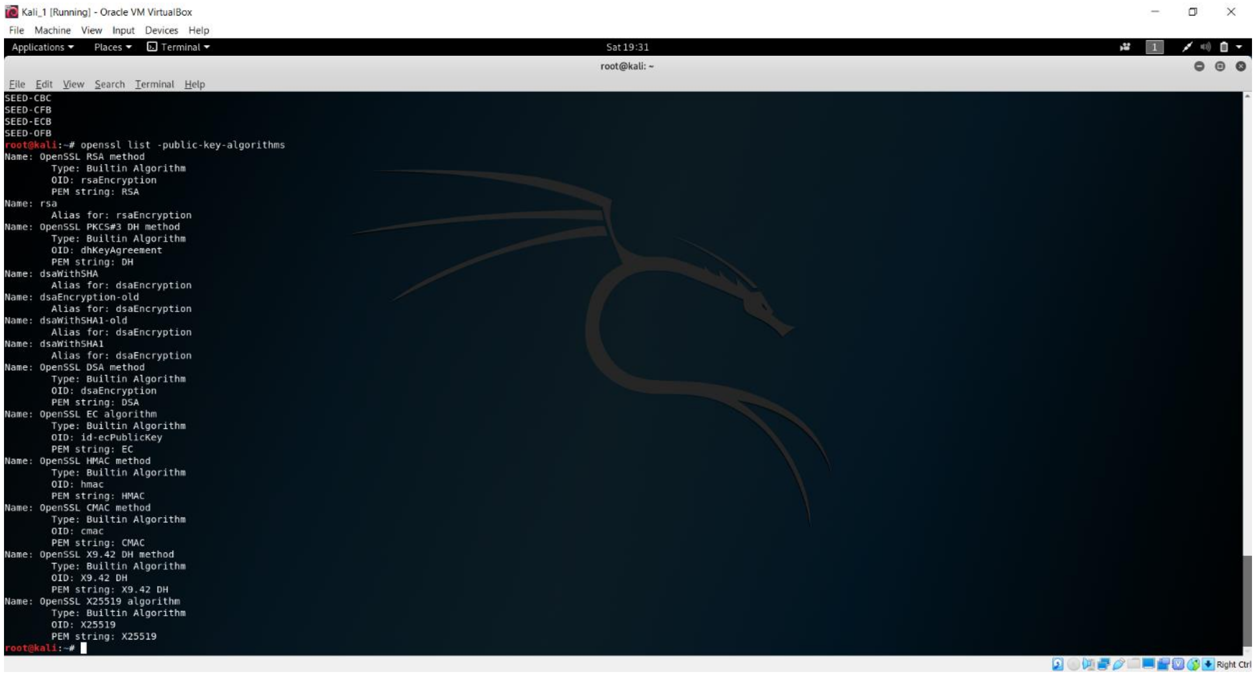
(2.2.6e1)



Kali_1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal
Sat 19:30
root@kali: ~
File Edit View Search Terminal Help
Right Ctrl

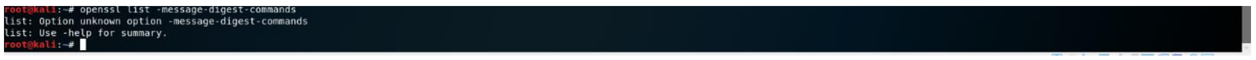
```
DES-EDE-CBC
DES-EDE-CFB
des-ede-ecb => DES-EDE
DES-EDE-OFB
DES-EDE3
DES-EDE3-CBC
DES-EDE3-CFB
DES-EDE3-CFB1
DES-EDE3-CFB8
des-ede3-ecb => DES-EDE3
DES-EDE3-OFB
DES-OFB
des3 => DES-EDE3-CBC
des3-wrap => id-smime-alg-CMS3DESwrap
desx => DESX-CBC
DESX-CBC
id-aes128-CCM
id-aes128-GCM
id-aes128-wrap
id-aes128-wrap-pad
id-aes192-CCM
id-aes192-GCM
id-aes192-wrap
id-aes192-wrap-pad
id-aes256-CCM
id-aes256-GCM
id-aes256-wrap
id-aes256-wrap-pad
id-smime-alg-CMS3DESwrap
rc2 => RC2-CBC
rc2-128 => RC2-CBC
rc2-40 => RC2-40-CBC
RC2-40-CBC
rc2-64 => RC2-64-CBC
RC2-64-CBC
RC2-CBC
RC2-CFB
RC2-ECB
RC2-OFB
RC4
RC4-40
RC4-HMAC-MD5
seed => SEED-CBC
SEED-CBC
SEED-CFB
SEED-ECB
SEED-OFB
root@kali:~# 
```

(2.2.6e2)


Kali_1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal
Sat 19:31
root@kali: ~
File Edit View Search Terminal Help
Right Ctrl

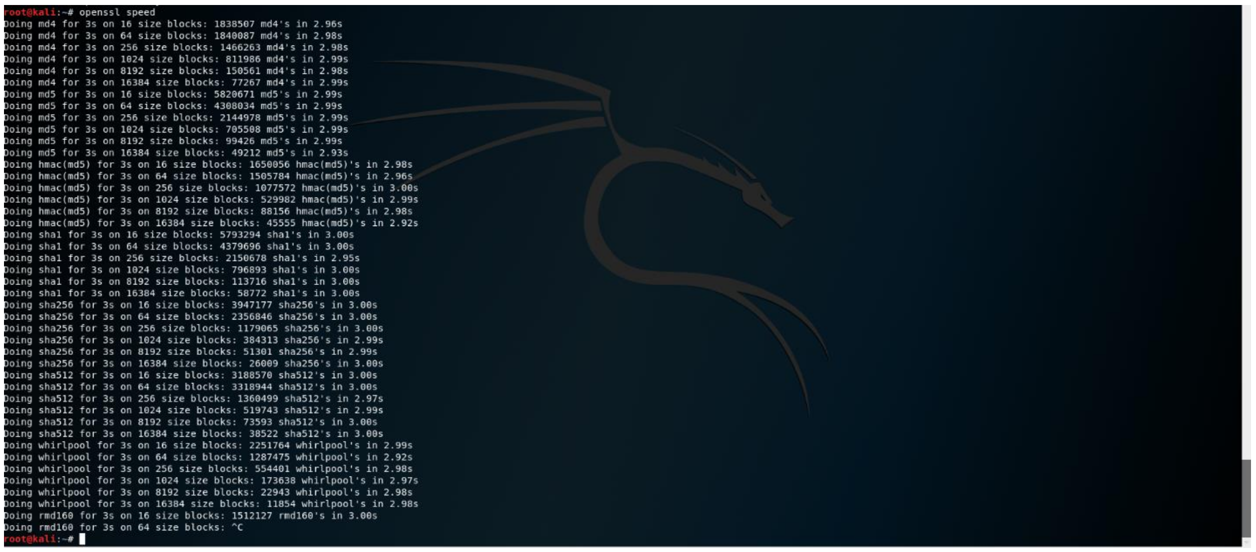
```
SEED-CBC
SEED-CFB
SEED-ECB
SEED-OFB
root@kali:~# openssl list -public-key-algorithms
Name: OpenSSL RSA method
        Type: Builtin Algorithm
        OID: rsaEncryption
        PEM string: RSA
Name: rsa
        Alias for: rsaEncryption
Name: OpenSSL PKCS#3 DH method
        Type: Builtin Algorithm
        OID: dhKeyAgreement
        PEM string: DH
Name: dsaWithSHA
        Alias for: dsaEncryption
Name: dsaEncryption-old
        Alias for: dsaEncryption
Name: dsaWithSHA1-old
        Alias for: dsaEncryption
Name: dsaWithSHA1
        Alias for: dsaEncryption
Name: OpenSSL DSA method
        Type: Builtin Algorithm
        OID: dsaEncryption
        PEM string: DSA
Name: OpenSSL EC algorithm
        Type: Builtin Algorithm
        OID: id-ecPublicKey
        PEM string: EC
Name: OpenSSL HMAC method
        Type: Builtin Algorithm
        OID: hmac
        PEM string: HMAC
Name: OpenSSL CMAC method
        Type: Builtin Algorithm
        OID: cmac
        PEM string: CMAC
Name: OpenSSL X9.42 DH method
        Type: Builtin Algorithm
        OID: X9.42 DH
        PEM string: X9.42 DH
Name: OpenSSL X25519 algorithm
        Type: Builtin Algorithm
        OID: X25519
        PEM string: X25519
root@kali:~#
```

(2.2.6e3)



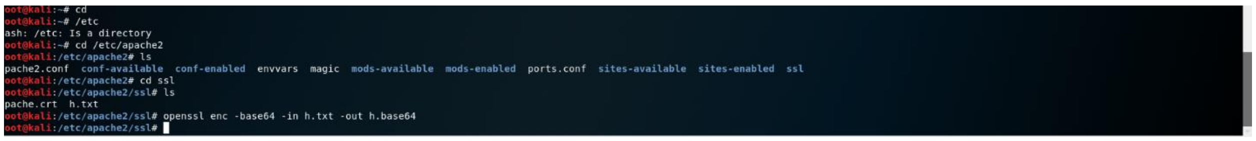
```
root@kali:~# openssl list -message-digest-commands
list: Option unknown option -message-digest-commands
list: Use -help for summary.
root@kali:~#
```

(2.2.6e4)



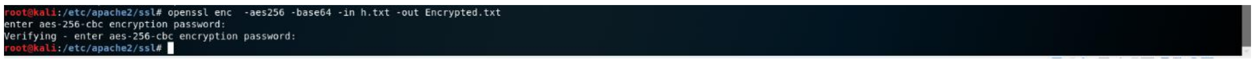
```
root@kali:~# openssl speed
Doing md4 for 3s on 16 size blocks: 1838507 md4's in 2.96s
Doing md4 for 3s on 64 size blocks: 1840087 md4's in 2.98s
Doing md4 for 3s on 256 size blocks: 1466263 md4's in 2.98s
Doing md4 for 3s on 1024 size blocks: 811986 md4's in 2.99s
Doing md4 for 3s on 8192 size blocks: 150561 md4's in 2.98s
Doing md4 for 3s on 16384 size blocks: 77267 md4's in 2.99s
Doing md5 for 3s on 16 size blocks: 5820671 md5's in 2.99s
Doing md5 for 3s on 64 size blocks: 4308034 md5's in 2.99s
Doing md5 for 3s on 256 size blocks: 2144978 md5's in 2.99s
Doing md5 for 3s on 1024 size blocks: 705508 md5's in 2.99s
Doing md5 for 3s on 8192 size blocks: 99426 md5's in 2.99s
Doing md5 for 3s on 16384 size blocks: 49212 md5's in 2.93s
Doing hmac(md5) for 3s on 16 size blocks: 1650056 hmac(md5)'s in 2.98s
Doing hmac(md5) for 3s on 64 size blocks: 1505784 hmac(md5)'s in 2.96s
Doing hmac(md5) for 3s on 256 size blocks: 1077572 hmac(md5)'s in 3.00s
Doing hmac(md5) for 3s on 1024 size blocks: 529982 hmac(md5)'s in 2.99s
Doing hmac(md5) for 3s on 8192 size blocks: 88156 hmac(md5)'s in 2.98s
Doing hmac(md5) for 3s on 16384 size blocks: 45555 hmac(md5)'s in 2.92s
Doing sha1 for 3s on 16 size blocks: 5793294 sha1's in 3.00s
Doing sha1 for 3s on 64 size blocks: 4379696 sha1's in 3.00s
Doing sha1 for 3s on 256 size blocks: 2150678 sha1's in 2.95s
Doing sha1 for 3s on 1024 size blocks: 796893 sha1's in 3.00s
Doing sha1 for 3s on 8192 size blocks: 113716 sha1's in 3.00s
Doing sha1 for 3s on 16384 size blocks: 58772 sha1's in 3.00s
Doing sha256 for 3s on 16 size blocks: 3947177 sha256's in 3.00s
Doing sha256 for 3s on 64 size blocks: 2356846 sha256's in 3.00s
Doing sha256 for 3s on 256 size blocks: 1179065 sha256's in 3.00s
Doing sha256 for 3s on 1024 size blocks: 384313 sha256's in 2.99s
Doing sha256 for 3s on 8192 size blocks: 51301 sha256's in 2.99s
Doing sha256 for 3s on 16384 size blocks: 26009 sha256's in 3.00s
Doing sha512 for 3s on 16 size blocks: 3188570 sha512's in 3.00s
Doing sha512 for 3s on 64 size blocks: 3318944 sha512's in 3.00s
Doing sha512 for 3s on 256 size blocks: 1360499 sha512's in 2.97s
Doing sha512 for 3s on 1024 size blocks: 519743 sha512's in 2.99s
Doing sha512 for 3s on 8192 size blocks: 73593 sha512's in 3.00s
Doing sha512 for 3s on 16384 size blocks: 38522 sha512's in 3.00s
Doing whirlpool for 3s on 16 size blocks: 2251764 whirlpool's in 2.99s
Doing whirlpool for 3s on 64 size blocks: 1287475 whirlpool's in 2.92s
Doing whirlpool for 3s on 256 size blocks: 554401 whirlpool's in 2.98s
Doing whirlpool for 3s on 1024 size blocks: 173638 whirlpool's in 2.97s
Doing whirlpool for 3s on 8192 size blocks: 22943 whirlpool's in 2.98s
Doing whirlpool for 3s on 16384 size blocks: 11854 whirlpool's in 2.98s
Doing rmd160 for 3s on 16 size blocks: 1512127 rmd160's in 3.00s
Doing rmd160 for 3s on 64 size blocks: ^C
root@kali:~#
```

(2.2.6e5)



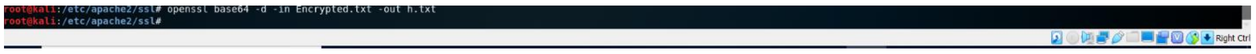
```
oot@kali:~# cd
oot@kali:~# /etc
ash: /etc: Is a directory
oot@kali:~# cd /etc/apache2
oot@kali:/etc/apache2# ls
pache2.conf  conf-available  conf-enabled  envvars  magic  mods-available  mods-enabled  ports.conf  sites-available  sites-enabled  ssl
oot@kali:/etc/apache2# cd ssl
oot@kali:/etc/apache2/ssl# ls
pache.crt  h.txt
oot@kali:/etc/apache2/ssl# openssl enc -base64 -in h.txt -out h.base64
oot@kali:/etc/apache2/ssl# 
```

(2.2.6e6)



```
root@kali:/etc/apache2/ssl# openssl enc  -aes256 -base64 -in h.txt -out Encrypted.txt
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
root@kali:/etc/apache2/ssl# 
```

(2.2.6e7)



```
root@kali:/etc/apache2/ssl# openssl base64 -d -in Encrypted.txt -out h.txt
root@kali:/etc/apache2/ssl#
```

Part 3:

1.1: Worm that caused denial of Service on some Internet Host and dramatically slowed down general internet traffic. It spread rapidly infecting most of 75k victims within 10 mins.

1.2: It infects new machines over User Datagram Protocol and program is small enough to fit inside a single packet.

1.3: The worm determines ip addresses of the victims host and subnets by generating random IP address and targeting another computer that could be anywhere on Internet. It also deploys time honored programmers trick by looking up number of milliseconds that have cpu time elapsed on CPU system clock

1.4: Functional requirement is OS must b windows and worm is written inx86 assembly and there must be vulnerability.

1.5: It communicates through the IP address

2. When the referenced capture is loaded and you look at frame 8 and 9 youll notice that it contains an IP fragment with payload of 36 bytes and next fragment would start at offset 36. If you look at frame 9 IP fragment starts at offset 24. Therefore this is an example of overlap which shows essence of teardrop attack.

The packet that causes the attack to be inaccurate is the use of “fragmentation feature”. In this case combination of IP fragment in frame 8 and frame 9 are the attack.

If you need to find source destination address you must look at the IP header of each fragments, even though the source address might be spoofed.