

## IP-SEC Tutorial

Objective: To see IP-SEC framework in action by establishing Host – To – Host IPSEC VPN Tunnel.

Required VMs: Can be found at `/virtual/ipsec/`

- VM1 | Centos 7
- VM2 | Centos 7
- Kali Linux | Kali Linux

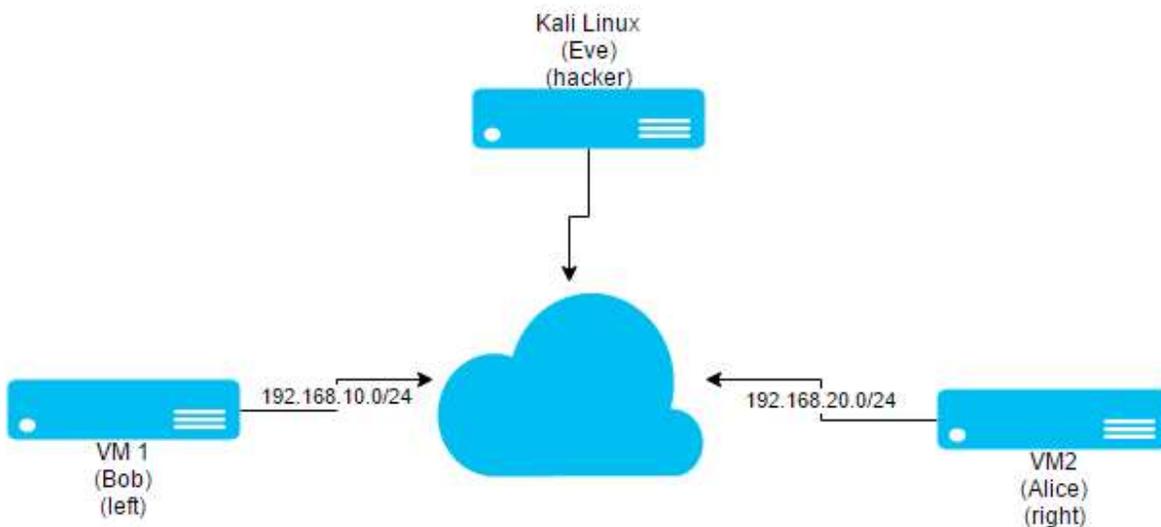
If you do not see the files use the following cmd:-

```
# scp -r [UTORID]@dh2020pc22.utm.utoronto.ca:/virtual/ipsec /virtual/ipsec
```

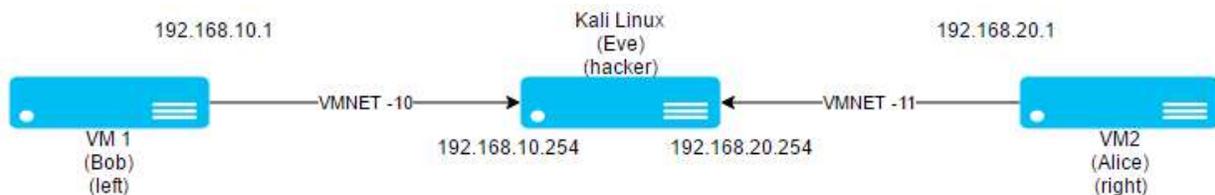
Required Software: Already installed for you on the VMs: -

- IPSEC Tools
- WireShark
- libreswan

### Scenario



How we emulate this: -



VM access details: -

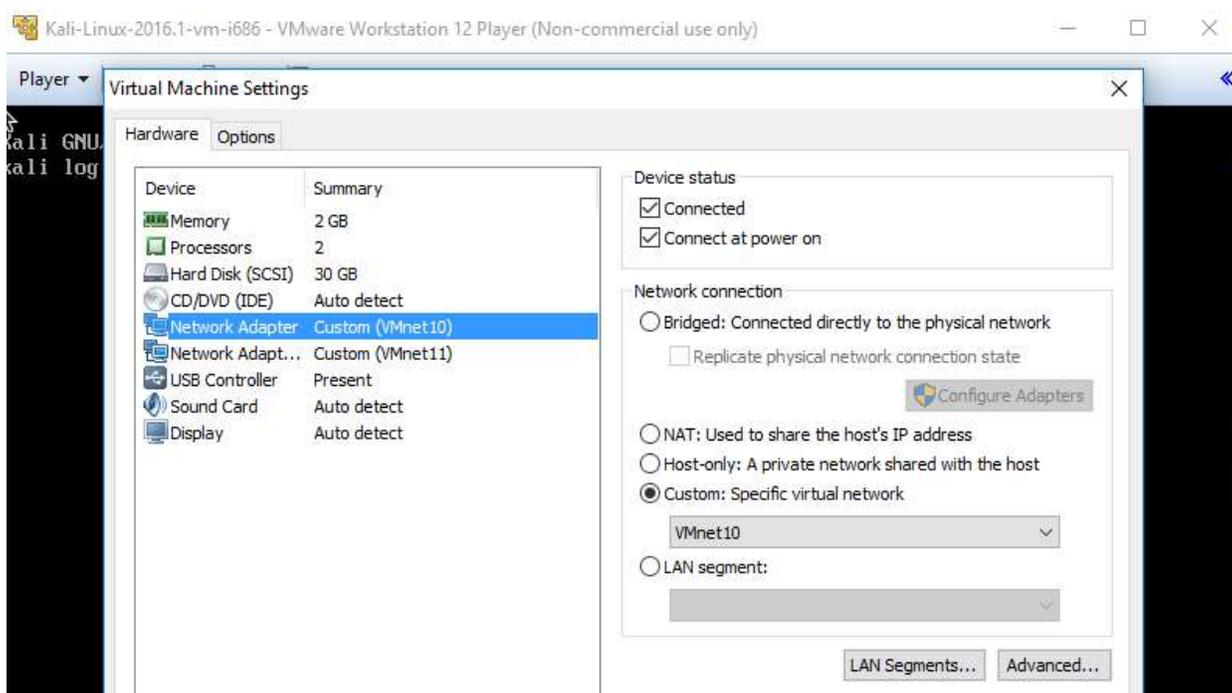
- Kali Linux: root/toor
- VM1: root/root
- VM2: root/root

SETUP: Please turn on all the VMs and login. For VM1 and VM2 you may use startx to use the terminal in GUI so that you can copy paste directly in the VMS as VMware tools is installed on all VMs.

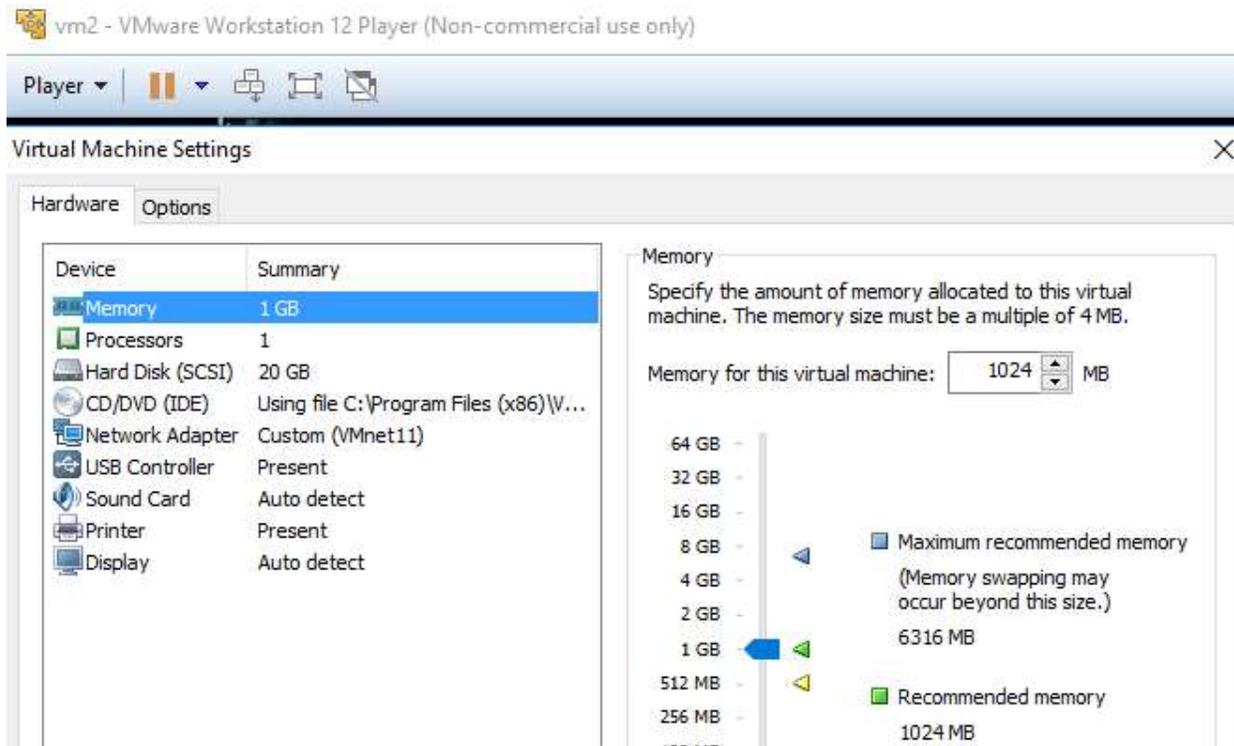
```
CentOS Linux 7 (Core)
Kernel 3.10.0-327.el7.x86_64 on an x86_64

vm2 login: root
Password:
Last login: Sun Mar 27 21:46:14 on tty1
[root@vm2 ~]#
[root@vm2 ~]#
[root@vm2 ~]# startx
```

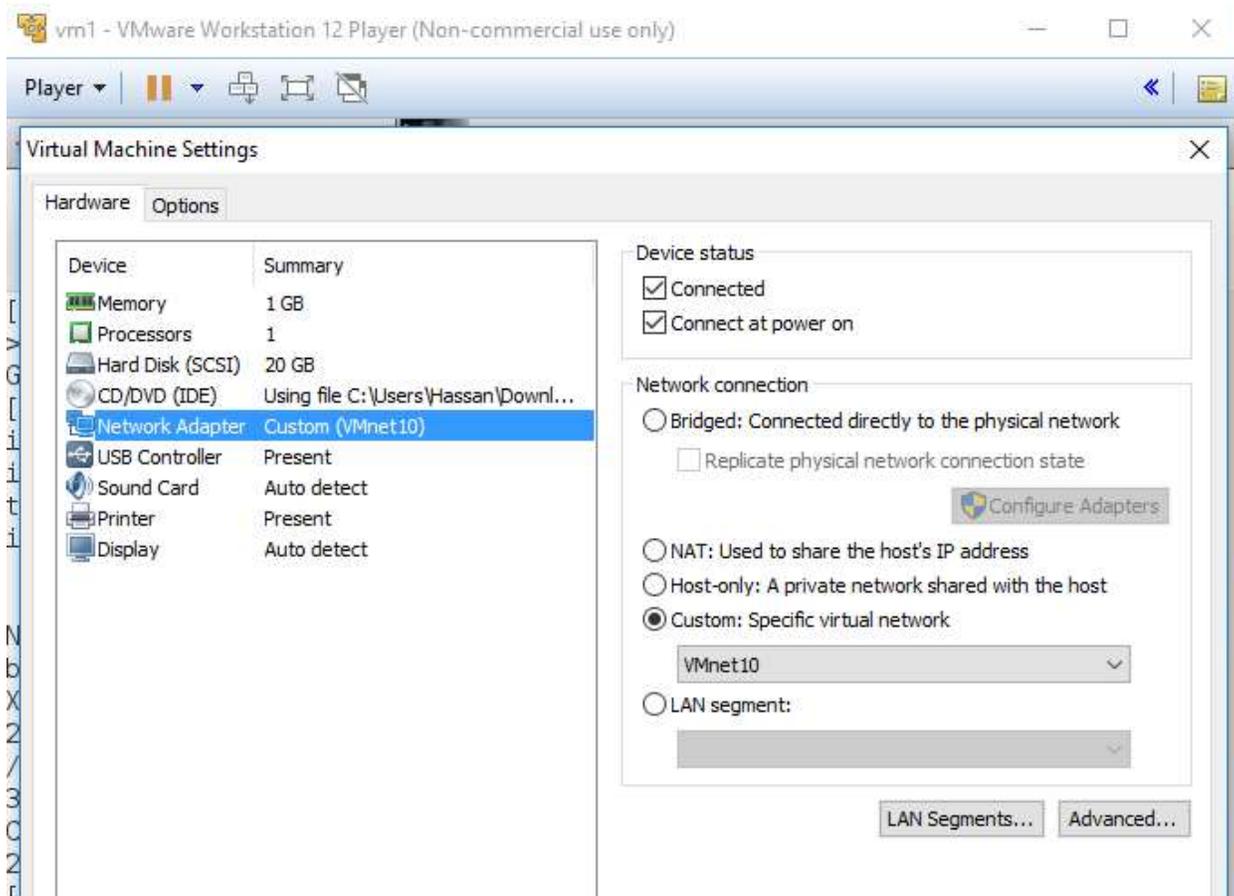
SETUP: Make sure that in Kali Linux VM Setting (the Network settings) are as below: -



SETUP: Make sure that in VM2 Setting (the Network settings) are as below: -



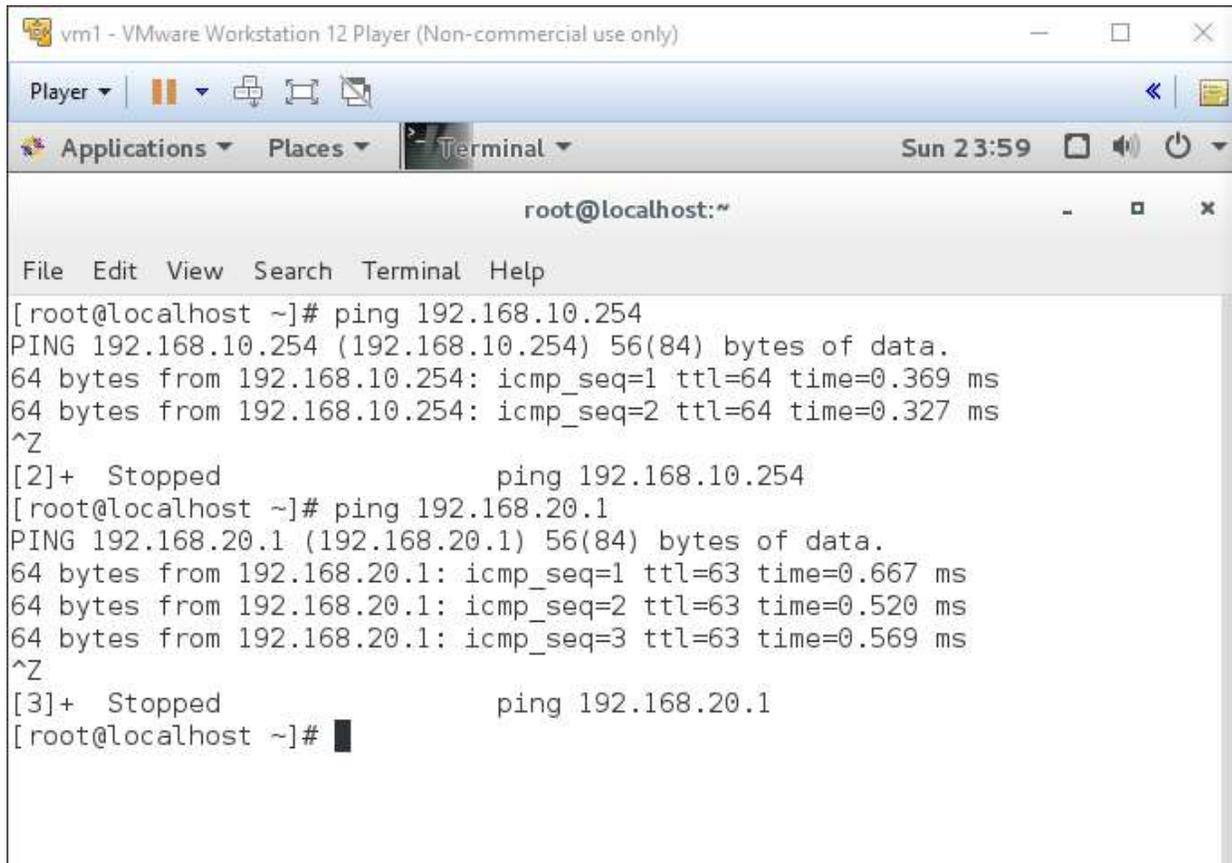
SETUP: Make sure that in Kali Linux VM1 Setting (the Network settings) are as below: -



SETUP: Make sure that all the VMs have the correct IPs as below:

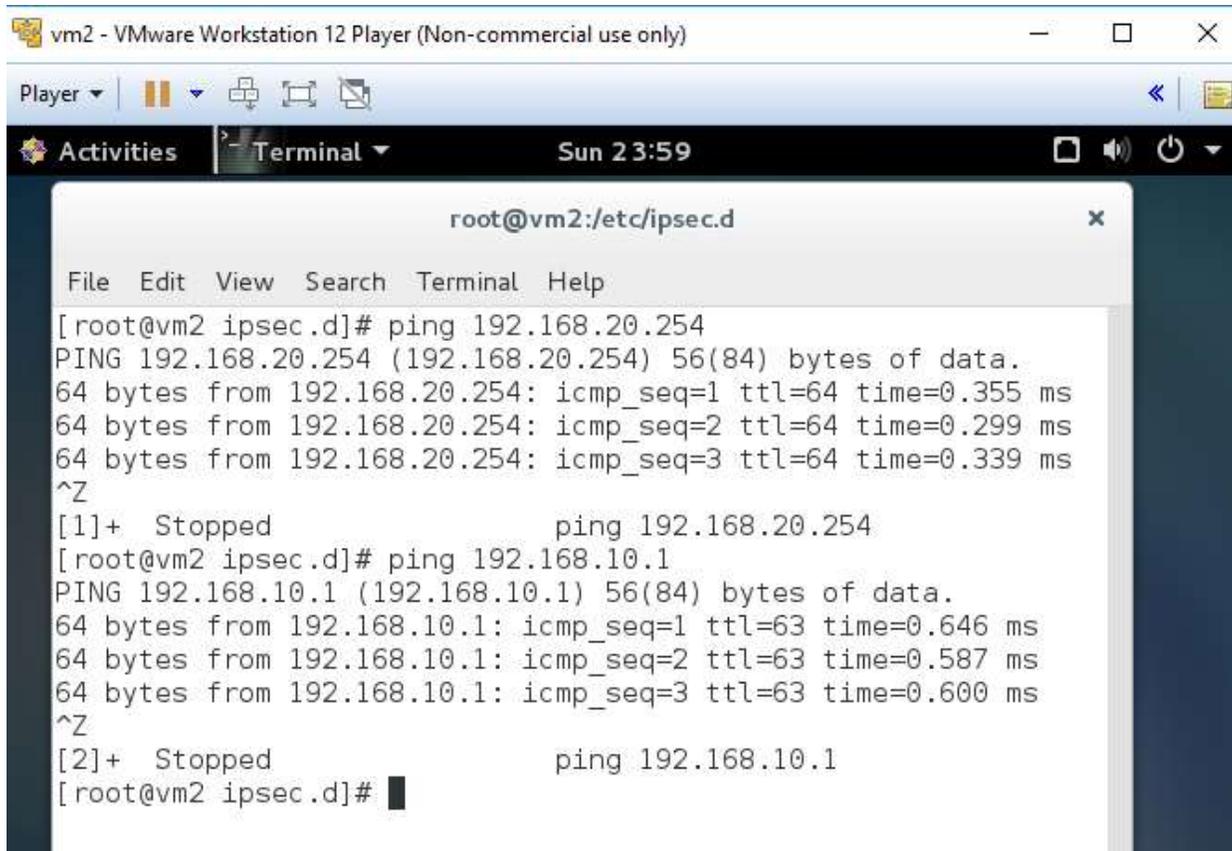
- VM1: 192.168.10.1 | 255.255.255.0 | 192.168.10.254
- VM2: 192.168.20.1 | 255.255.255.0 | 192.168.20.254
- Kali Linux: eth1: 192.168.20.254 | 255.255.255.0
- Kali Linux: eth0: 192.168.10.254 | 255.255.255.0

Ensure that you can ping the other VMs from all VMs as per below:-

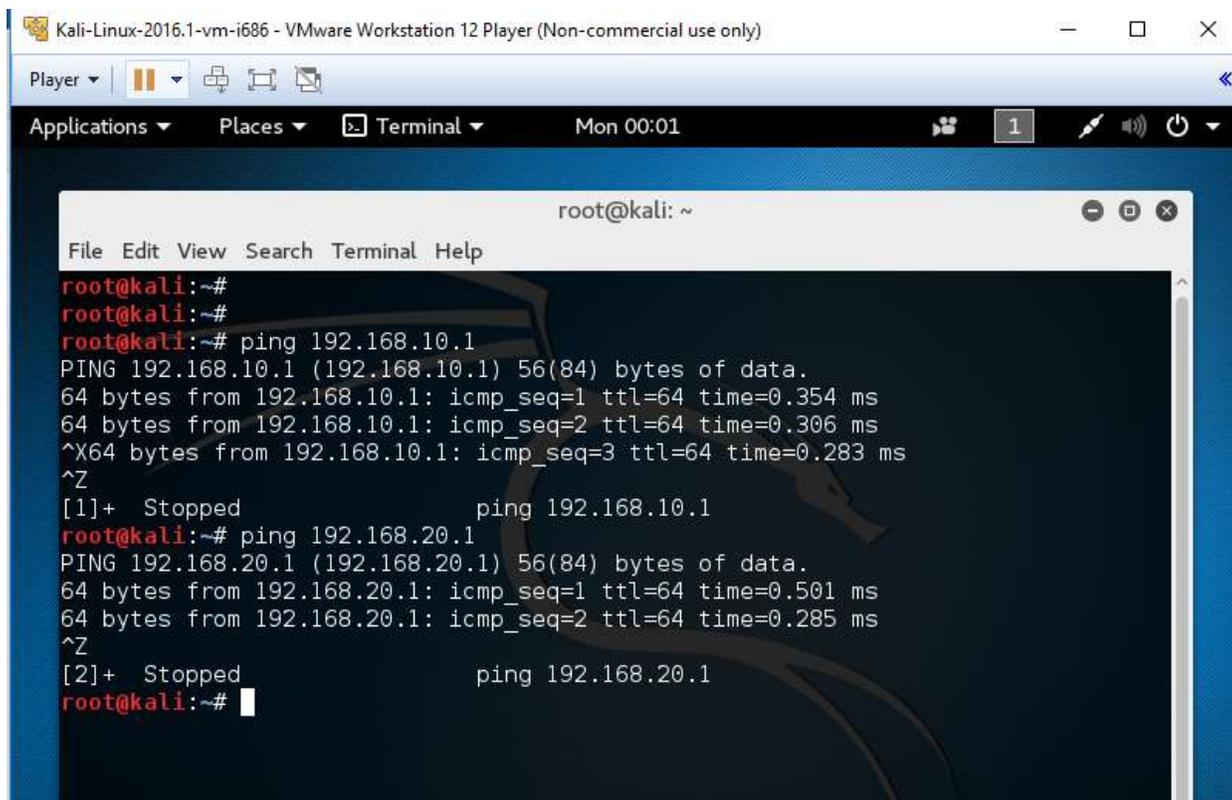


The screenshot shows a terminal window titled "vm1 - VMware Workstation 12 Player (Non-commercial use only)". The terminal prompt is "root@localhost:~". The user has performed two ping tests. The first test is for 192.168.10.254, showing successful results with TTL=64 and times around 0.3 ms. The second test is for 192.168.20.1, also showing successful results with TTL=63 and times around 0.5 ms. The terminal output is as follows:

```
root@localhost:~# ping 192.168.10.254
PING 192.168.10.254 (192.168.10.254) 56(84) bytes of data.
64 bytes from 192.168.10.254: icmp_seq=1 ttl=64 time=0.369 ms
64 bytes from 192.168.10.254: icmp_seq=2 ttl=64 time=0.327 ms
^Z
[2]+  Stopped                  ping 192.168.10.254
root@localhost ~]# ping 192.168.20.1
PING 192.168.20.1 (192.168.20.1) 56(84) bytes of data.
64 bytes from 192.168.20.1: icmp_seq=1 ttl=63 time=0.667 ms
64 bytes from 192.168.20.1: icmp_seq=2 ttl=63 time=0.520 ms
64 bytes from 192.168.20.1: icmp_seq=3 ttl=63 time=0.569 ms
^Z
[3]+  Stopped                  ping 192.168.20.1
root@localhost ~]# █
```

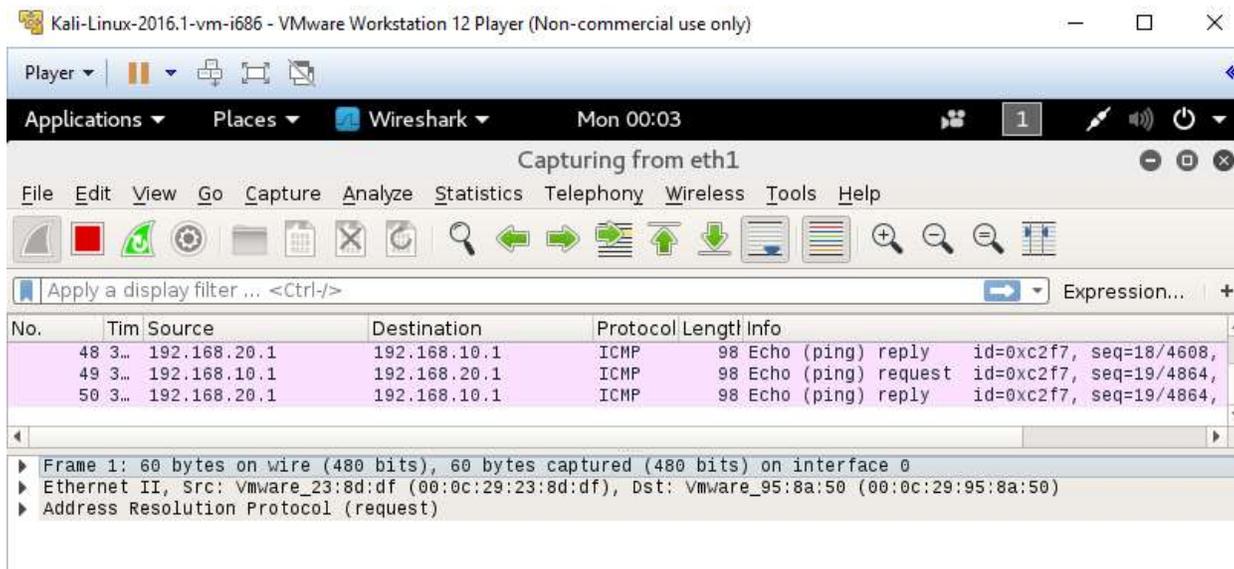


```
vm2 - VMware Workstation 12 Player (Non-commercial use only)
Player
Activities Terminal Sun 23:59
root@vm2:/etc/ipsec.d
File Edit View Search Terminal Help
[root@vm2 ipsec.d]# ping 192.168.20.254
PING 192.168.20.254 (192.168.20.254) 56(84) bytes of data.
64 bytes from 192.168.20.254: icmp_seq=1 ttl=64 time=0.355 ms
64 bytes from 192.168.20.254: icmp_seq=2 ttl=64 time=0.299 ms
64 bytes from 192.168.20.254: icmp_seq=3 ttl=64 time=0.339 ms
^Z
[1]+  Stopped                  ping 192.168.20.254
[root@vm2 ipsec.d]# ping 192.168.10.1
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data.
64 bytes from 192.168.10.1: icmp_seq=1 ttl=63 time=0.646 ms
64 bytes from 192.168.10.1: icmp_seq=2 ttl=63 time=0.587 ms
64 bytes from 192.168.10.1: icmp_seq=3 ttl=63 time=0.600 ms
^Z
[2]+  Stopped                  ping 192.168.10.1
[root@vm2 ipsec.d]#
```

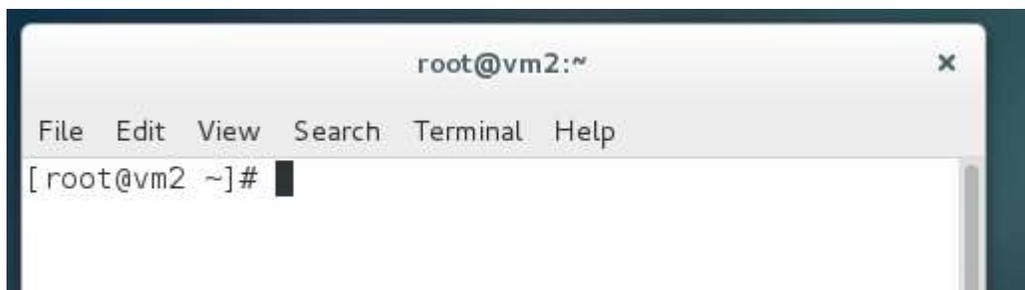


```
Kali-Linux-2016.1-vm-i686 - VMware Workstation 12 Player (Non-commercial use only)
Player
Applications Places Terminal Mon 00:01
root@kali: ~
File Edit View Search Terminal Help
root@kali:~#
root@kali:~#
root@kali:~# ping 192.168.10.1
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data.
64 bytes from 192.168.10.1: icmp_seq=1 ttl=64 time=0.354 ms
64 bytes from 192.168.10.1: icmp_seq=2 ttl=64 time=0.306 ms
^X64 bytes from 192.168.10.1: icmp_seq=3 ttl=64 time=0.283 ms
^Z
[1]+  Stopped                  ping 192.168.10.1
root@kali:~# ping 192.168.20.1
PING 192.168.20.1 (192.168.20.1) 56(84) bytes of data.
64 bytes from 192.168.20.1: icmp_seq=1 ttl=64 time=0.501 ms
64 bytes from 192.168.20.1: icmp_seq=2 ttl=64 time=0.285 ms
^Z
[2]+  Stopped                  ping 192.168.20.1
root@kali:~#
```

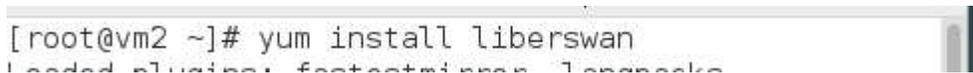
SETUP: Open Wireshark in Kali Linux. Monitor eth1 and see if you can see the ping traffic pass through the Kali Linux.



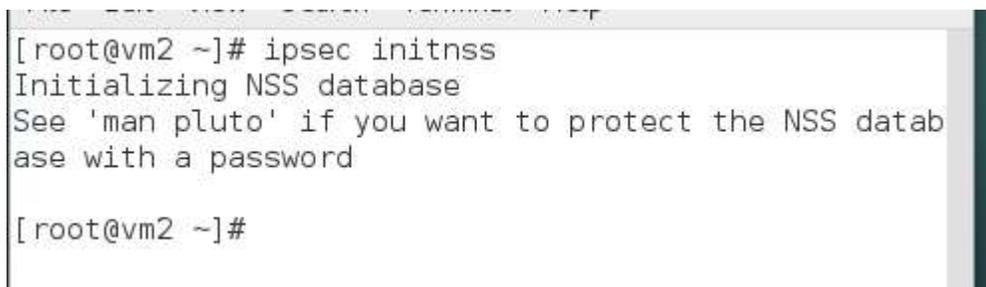
STEP: Start the terminal in VM1 and VM2.



STEP: Install Libreswan (already done for you) in both VMs.



Step: Initialize a new database. If a db already exist use `rm /etc/ipsec.d/*db` to remove and then initialize a new one on both VMs.



STEP: Check if IPSEC service is running on both VMs.

```

root@vm2:~
File Edit View Search Terminal Help
[root@vm2 ~]# systemctl status ipsec
● ipsec.service - Internet Key Exchange (IKE) Protocol Daemon for IPsec
   Loaded: loaded (/usr/lib/systemd/system/ipsec.service; disabled; vendor preset: disabled)
   Active: inactive (dead)
[root@vm2 ~]# █

```

STEP: Start IPSEC Service on both VMs.

```

[root@vm2 ~]# systemctl start ipsec
[root@vm2 ~]# █

```

STEP: Check the status again on both VMs.

```

[root@vm2 ~]# systemctl status ipsec
● ipsec.service - Internet Key Exchange (IKE) Protocol Daemon for IPsec
   Loaded: loaded (/usr/lib/systemd/system/ipsec.service; disabled; vendor preset: disabled)
   Active: active (running) since Sun 2016-03-27 23:27:49 EDT; 29s ago
     Process: 4638 ExecStartPre=/usr/sbin/ipsec --check-nflog (code=exited, status=0/SUCCESS)
     Process: 4633 ExecStartPre=/usr/sbin/ipsec --check-nss (code=exited, status=0/SUCCESS)

```

STEP: Important to add IPSEC to start on startup on both VMs.

```

[root@vm2 ~]# systemctl enable ipsec
Created symlink from /etc/systemd/system/multi-user.target.wants/ipsec.service to /usr/lib/systemd/system/ipsec.service.
[root@vm2 ~]# █

```

### ! IMPORTANT INFO !

**We are implementing HOST – to – HOST IPSEC VPN Tunnel**

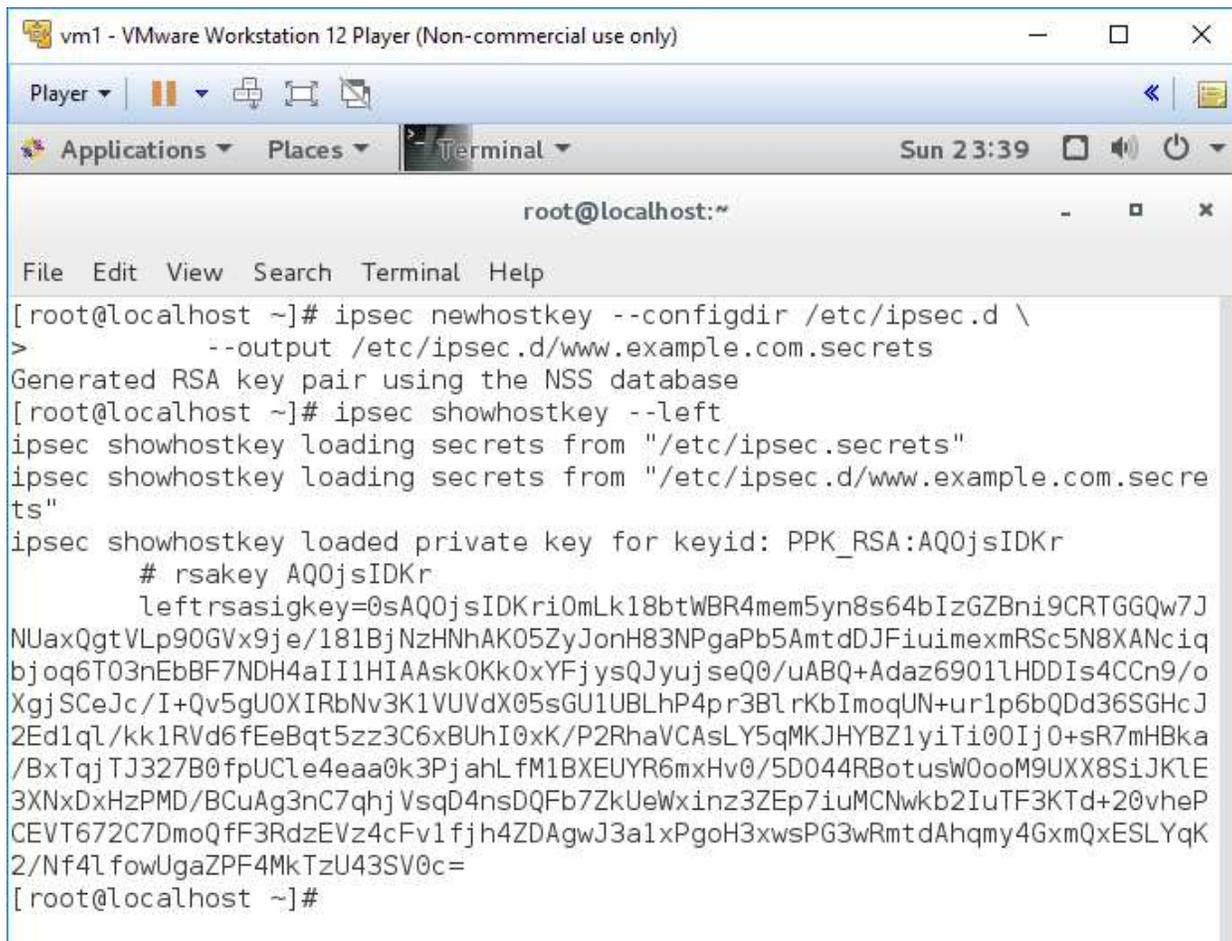
**The two hosts are referred to as “left” and “right”.**

**We are going to use vm1 as the “left”.**

**And vm2 as “right”.**

STEP: Generate an rsa key for VM1 (left) as per below: -

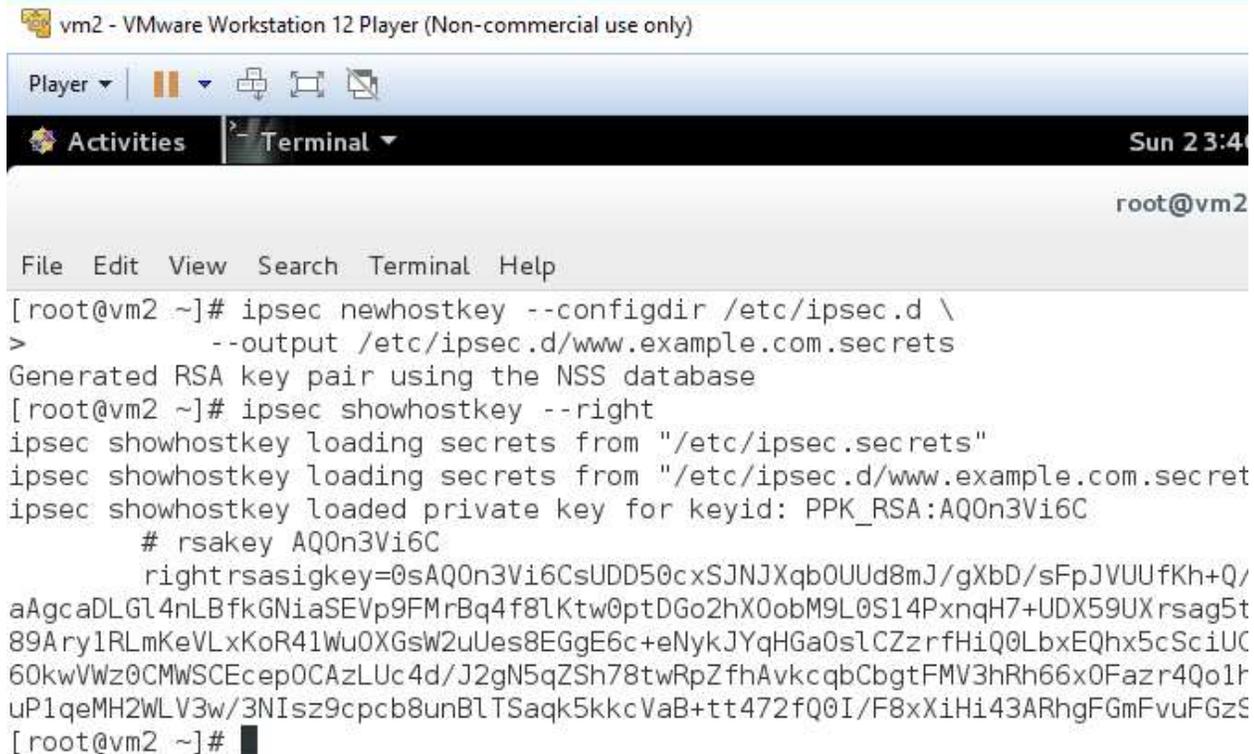
```
# ipsec newhostkey --configdir /etc/ipsec.d \  
    --output /etc/ipsec.d/www.example.com.secrets  
  
# ipsec showhostkey --left
```



```
vm1 - VMware Workstation 12 Player (Non-commercial use only)  
Player | [Icons] | [Navigation]  
Applications | Places | Terminal | Sun 23:39 | [System Icons]  
root@localhost:~  
File Edit View Search Terminal Help  
[root@localhost ~]# ipsec newhostkey --configdir /etc/ipsec.d \  
> --output /etc/ipsec.d/www.example.com.secrets  
Generated RSA key pair using the NSS database  
[root@localhost ~]# ipsec showhostkey --left  
ipsec showhostkey loading secrets from "/etc/ipsec.secrets"  
ipsec showhostkey loading secrets from "/etc/ipsec.d/www.example.com.secrets"  
ipsec showhostkey loaded private key for keyid: PPK_RSA:AQ0jsIDKr  
# rsakey AQ0jsIDKr  
leftrsasigkey=0sAQ0jsIDKri0mLk18btWBR4mem5yn8s64bIzGZBni9CRTGGQw7J  
NUaxQgtVLp90GVx9je/181BjNzHNhAK05ZyJonH83NPgaPb5AmdtDJFiuimexmRSc5N8XANciq  
bjoq6T03nEbBF7NDH4aII1HIAAsk0Kk0xYFjysQJyujseQ0/uABQ+Adaz6901lHDDIs4CCn9/o  
XgJSceJc/I+Qv5gUOXIRbNv3K1VUVdX05sGU1UBLhP4pr3B1rKbImoqUN+ur1p6bQDd36SGHcJ  
2Ed1qL/kk1RVd6fEeBqt5zz3C6xBUhI0xK/P2RhaVCAAsLY5qMKJHYBZ1yiTi00Ij0+sR7mHBka  
/BxTqjTJ327B0fpUCLe4eaa0k3PjahL fM1BXEUyR6mxHv0/5D044RBotusW0ooM9UXX8SiJKLE  
3XNx DxHzPMD/BCuAg3nC7qhjVsqD4nsDQFb7ZkUeWxinz3ZEp7iuMCNwkb2IuTF3KTd+20vheP  
CEVT672C7DmoQfF3RdzEVz4cFv1fjh4ZDAgwJ3a1xPgoH3xwsPG3wRmtdAhqmy4GxmQxESLYqK  
2/Nf4lfowUgaZPF4MkTzU43SV0c=  
[root@localhost ~]#
```

STEP: Generate rsa key for VM2 as per below: -

```
# ipsec newhostkey --configdir /etc/ipsec.d \
    --output /etc/ipsec.d/www.example.com.secrets
# ipsec showhostkey --right
```



```
vm2 - VMware Workstation 12 Player (Non-commercial use only)
Player
Activities Terminal Sun 23:4
root@vm2
File Edit View Search Terminal Help
[root@vm2 ~]# ipsec newhostkey --configdir /etc/ipsec.d \
> --output /etc/ipsec.d/www.example.com.secrets
Generated RSA key pair using the NSS database
[root@vm2 ~]# ipsec showhostkey --right
ipsec showhostkey loading secrets from "/etc/ipsec.secrets"
ipsec showhostkey loading secrets from "/etc/ipsec.d/www.example.com.secret
ipsec showhostkey loaded private key for keyid: PPK_RSA:AQ0n3Vi6C
# rsakey AQ0n3Vi6C
rightrsasigkey=0sAQ0n3Vi6CsUDD50cxSjNjXqb0Uud8mJ/gXbD/sFpJVUUFKh+Q/
aAgcaDLG14nLBfkGNiaSEVp9FMrBq4f8lKtw0ptDGo2hX0obM9L0S14PxnqH7+UDX59UXrsag5t
89Ary1RLmKeVLxKoR41Wu0XGsw2uUes8EGgE6c+eNykJYqHGa0slCZzrfHiQ0LbxEQhx5cSciUC
60kwVWz0CMWSCEcep0CAzLUc4d/J2gN5qZSh78twRpZfhAvkcbCbgtFMV3hRh66x0Fazr4Qo1f
uP1qeMH2WLV3w/3NIsz9cpcb8unBLTSaQk5kkcVaB+tt472fQ0I/F8xXiHi43ARhgFGmFvuFGzS
[root@vm2 ~]# █
```

STEP: Create a new IPSEC config file using for favorite editor in both VMs: -

```
# nano /etc/ipsec.d/my_host-to-host.conf
```

```
[root@vm2 ipsec.d]# nano /etc/ipsec.d/my_host-to-host.conf
[root@vm2 ipsec.d]# █
```

STEP: In the config file we can place the IPSEC configuration info as below in both VMs: -

```
conn mytunnel
    leftid=@west.example.com
    left=192.1.2.23
    leftrsasigkey={Paste rsa key for left as generate above}
    rightid=@east.example.com
    right=192.1.2.45
    rightrsasigkey={Paste rsa key for right as generate above}
    authby=rsasig
    # load and initiate automatically
    auto=start
```

vm2 - VMware Workstation 12 Player (Non-commercial use only)

Player ▾ | 

Activities | Terminal ▾ Sun 23:5

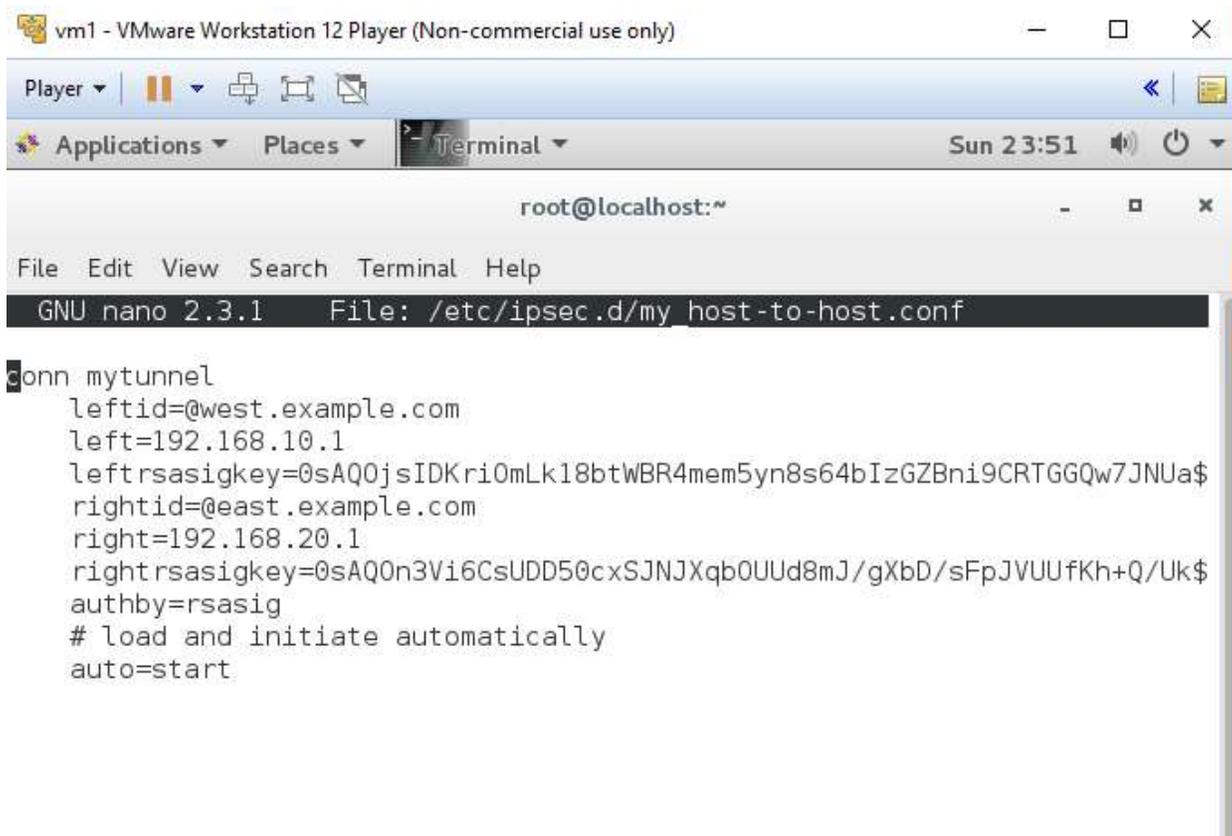
root@vm2:/etc/ipsec.d

File Edit View Search Terminal Help

GNU nano 2.3.1 File: ...sec.d/my\_host-to-host.conf Modified

```
conn mytunnel
    leftid=@west.example.com
    left=192.168.10.1
    leftrsasigkey=0sAQ0jsIDKri0mLk18btWBR4mem5yn8s64bIzGZBni9C$
    rightid=@east.example.com
    right=192.168.20.1
    rightrsasigkey=0sAQ0n3Vi6CsUDD50cxSJNJXqb0UUd8mJ/gXbD/sFpJ$
    authby=rsasig
    # load and initiate automatically
    auto=start
```

**IMPORTANT:** Ensure that both VMs have the same config file. They must be identical otherwise the SA's will fail.



```
vm1 - VMware Workstation 12 Player (Non-commercial use only)
Player
Applications Places Terminal Sun 23:51
root@localhost:~
File Edit View Search Terminal Help
GNU nano 2.3.1 File: /etc/ipsec.d/my_host-to-host.conf

conn mytunnel
leftid=@west.example.com
left=192.168.10.1
leftrsasigkey=@sAQ0jsIDKri0mLk18btWBR4mem5yn8s64bIzGZBni9CRTGGQw7JNUa$
rightid=@east.example.com
right=192.168.20.1
rightrsasigkey=@sAQ0n3Vi6CsUDD50cxSJNjXqb0UUd8mJ/gXbD/sFpJVUUFKh+Q/Uk$
authby=rsasig
# load and initiate automatically
auto=start
```

**BONUS:** Sometimes we need to ensure that libreswan can find our custom config file. Therefore, we must ensure that the following is commented out: -

```
# nano /etc/ipsec.conf
```

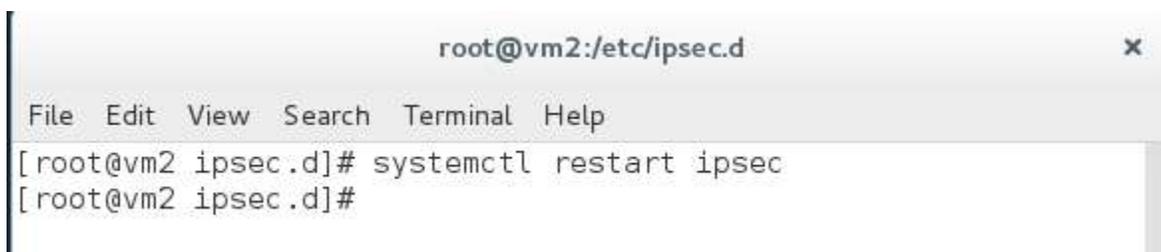
And Ensure that (should be at the end of the main config file): -

```
include /etc/ipsec.d/*.conf
```

is not commented out.

**STEP:** Restart the IPSEC service on both VMs so that they can pick up the config file that we created.

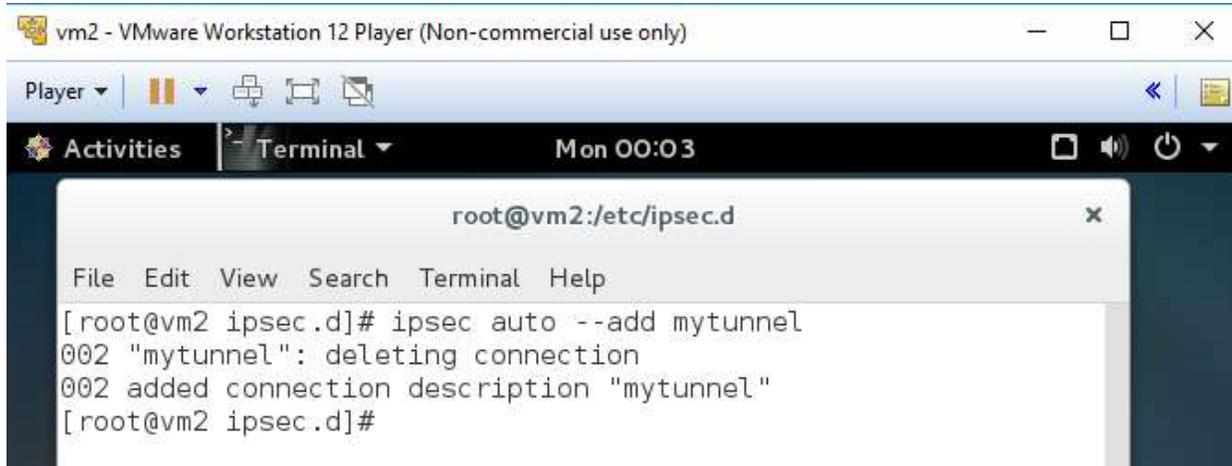
```
# systemctl restart ipsec
```



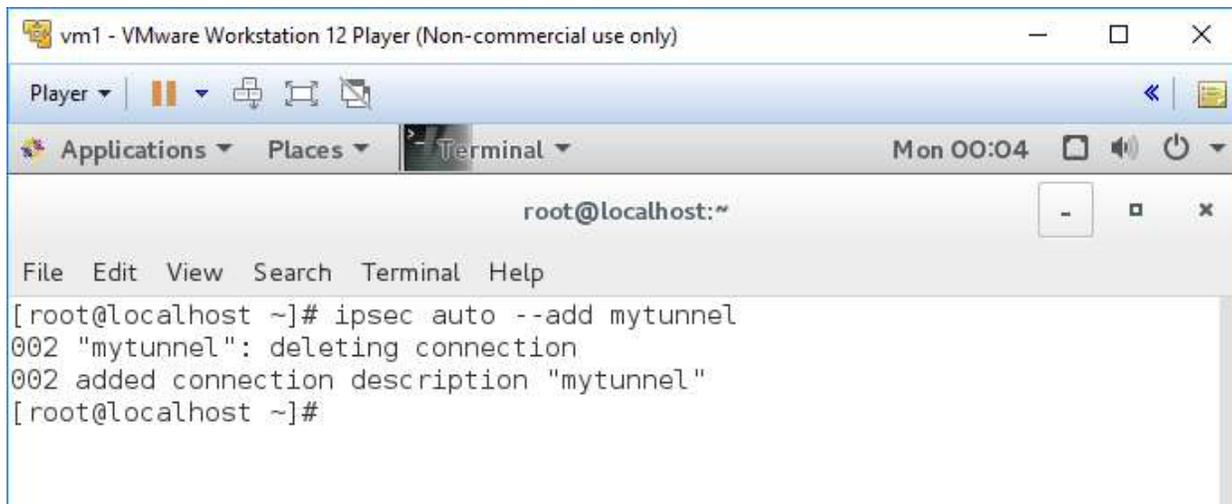
```
root@vm2:/etc/ipsec.d
File Edit View Search Terminal Help
[root@vm2 ipsec.d]# systemctl restart ipsec
[root@vm2 ipsec.d]#
```

STEP: On both VMs add the tunnel that we created: -

```
# ipsec auto --add mytunnel
```



```
vm2 - VMware Workstation 12 Player (Non-commercial use only)
Player
Activities Terminal Mon 00:03
root@vm2:/etc/ipsec.d
File Edit View Search Terminal Help
[root@vm2 ipsec.d]# ipsec auto --add mytunnel
002 "mytunnel": deleting connection
002 added connection description "mytunnel"
[root@vm2 ipsec.d]#
```



```
vm1 - VMware Workstation 12 Player (Non-commercial use only)
Player
Applications Places Terminal Mon 00:04
root@localhost:~
File Edit View Search Terminal Help
[root@localhost ~]# ipsec auto --add mytunnel
002 "mytunnel": deleting connection
002 added connection description "mytunnel"
[root@localhost ~]#
```

STEP: On any "ONE" of the VM turn on the connection: -

```
# ipsec auto --up mytunnel
```

```
[root@vm2 ipsec.d]# ipsec auto --up mytunnel
002 "mytunnel" #4: initiating Main Mode
104 "mytunnel" #4: STATE_MAIN_I1: initiate
003 "mytunnel" #4: received Vendor ID payload [Dead Peer Detect
ion]
003 "mytunnel" #4: received Vendor ID payload [FRAGMENTATION]
003 "mytunnel" #4: received Vendor ID payload [RFC 3947]
002 "mytunnel" #4: enabling possible NAT-traversal with method
RFC 3947 (NAT-Traversal)
002 "mytunnel" #4: transition from state STATE_MAIN_I1 to state
STATE_MAIN_I2
106 "mytunnel" #4: STATE_MAIN_I2: sent MI2, expecting MR2
003 "mytunnel" #4: NAT-Traversal: Result using RFC 3947 (NAT-Tr
aversal) sender port 500: no NAT detected
002 "mytunnel" #4: transition from state STATE_MAIN_I2 to state
STATE_MAIN_I3
108 "mytunnel" #4: STATE_MAIN_I3: sent MI3, expecting MR3
003 "mytunnel" #4: received Vendor ID payload [CAN-IKEv2]
003 "mytunnel" #4: Main mode peer ID is ID_500M: local success
```

IMPORTANT: In Wireshark on Kali Linux you will see ISAKMP connection packets establishing the IPSEC tunnel. Please ensure that you stop once you see these packets and explore them.

Kali-Linux-2016.1-vm-i686 - VMware Workstation 12 Player (Non-commercial use only)

Player | Applications | Places | Wireshark | Mon 00:12

\*eth1

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression...

| No. | Time | Source          | Destination     | Protocol | Length | Info                                      |
|-----|------|-----------------|-----------------|----------|--------|---|
| 119 | 5... | Vmware_95:8a:50 | Vmware_23:8d:df | ARP      | 42     | 192.168.20.254 is at 00:0c:29:95:8a:50    |
| 120 | 5... | 192.168.20.1    | 192.168.10.1    | ISAKMP   | 834    | Identity Protection (Main Mode)           |
| 121 | 5... | 192.168.10.1    | 192.168.20.1    | ISAKMP   | 186    | Identity Protection (Main Mode)           |
| 122 | 5... | 192.168.20.1    | 192.168.10.1    | ISAKMP   | 398    | Identity Protection (Main Mode)           |
| 123 | 5... | 192.168.10.1    | 192.168.20.1    | ISAKMP   | 398    | Identity Protection (Main Mode)           |
| 124 | 5... | 192.168.20.1    | 192.168.10.1    | ISAKMP   | 598    | Identity Protection (Main Mode)           |
| 125 | 5... | 192.168.10.1    | 192.168.20.1    | ISAKMP   | 566    | Identity Protection (Main Mode)           |
| 126 | 5... | 192.168.20.1    | 192.168.10.1    | ISAKMP   | 518    | Quick Mode                                |
| 127 | 5... | 192.168.10.1    | 192.168.20.1    | ISAKMP   | 438    | Quick Mode                                |
| 128 | 5... | 192.168.20.1    | 192.168.10.1    | ISAKMP   | 102    | Quick Mode                                |
| 129 | 5... | Vmware_23:8d:df | Vmware_95:8a:50 | ARP      | 60     | Who has 192.168.20.254? Tell 192.168.20.1 |
| 130 | 5... | Vmware_95:8a:50 | Vmware_23:8d:df | ARP      | 42     | 192.168.20.254 is at 00:0c:29:95:8a:50    |
| 131 | 5... | 192.168.10.1    | 192.168.20.1    | ISAKMP   | 518    | Quick Mode                                |
| 132 | 5... | 192.168.20.1    | 192.168.10.1    | ISAKMP   | 438    | Quick Mode                                |
| 133 | 5... | 192.168.10.1    | 192.168.20.1    | ISAKMP   | 102    | Quick Mode                                |
| 134 | 5... | Vmware_95:8a:50 | Vmware_23:8d:df | ARP      | 42     | Who has 192.168.20.1? Tell 192.168.20.254 |
| 135 | 5... | Vmware_23:8d:df | Vmware_95:8a:50 | ARP      | 60     | 192.168.20.1 is at 00:0c:29:23:8d:df      |

Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0

- Ethernet II, Src: Vmware\_23:8d:df (00:0c:29:23:8d:df), Dst: Vmware\_95:8a:50 (00:0c:29:95:8a:50)
- Address Resolution Protocol (request)

```

0000  00 0c 29 95 8a 50 00 0c 29 23 8d df 08 06 00 01  ..P..)#.....
0010  08 00 06 04 00 01 00 0c 29 23 8d df c0 a8 14 01  .....)#.....
0020  00 00 00 00 00 00 c0 a8 14 fe 00 00 00 00 00 00  .....
0030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
    
```

STEP: Ping from either VM1 or VM2 and you can observe that in Kali linux you can see the ESP encrypted packets:

Kali-Linux-2016.1-vm-i686 - VMware Workstation 12 Player (Non-commercial use only)

Applications | Places | Wireshark | Mon 00:12

Capturing from eth1

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression... +

| No. | Time | Source       | Destination  | Protocol | Length | Info                 |
|-----|------|--------------|--------------|----------|--------|----------------------|
| 1   | 0... | 192.168.10.1 | 192.168.20.1 | ESP      | 166    | ESP (SPI=0x7179cf80) |
| 2   | 0... | 192.168.20.1 | 192.168.10.1 | ESP      | 166    | ESP (SPI=0xe63f80ae) |
| 3   | 1... | 192.168.10.1 | 192.168.20.1 | ESP      | 166    | ESP (SPI=0x7179cf80) |
| 4   | 1... | 192.168.20.1 | 192.168.10.1 | ESP      | 166    | ESP (SPI=0xe63f80ae) |
| 5   | 2... | 192.168.10.1 | 192.168.20.1 | ESP      | 166    | ESP (SPI=0x7179cf80) |
| 6   | 2... | 192.168.20.1 | 192.168.10.1 | ESP      | 166    | ESP (SPI=0xe63f80ae) |
| 7   | 3... | 192.168.10.1 | 192.168.20.1 | ESP      | 166    | ESP (SPI=0x7179cf80) |
| 8   | 3... | 192.168.20.1 | 192.168.10.1 | ESP      | 166    | ESP (SPI=0xe63f80ae) |

▶ Frame 1: 166 bytes on wire (1328 bits), 166 bytes captured (1328 bits) on interface 0  
 ▶ Ethernet II, Src: VMware\_95:8a:50 (00:0c:29:95:8a:50), Dst: VMware\_23:8d:df (00:0c:29:23:8d:df)  
 ▶ Internet Protocol Version 4, Src: 192.168.10.1, Dst: 192.168.20.1  
 ▶ Encapsulating Security Payload

```

0000  00 0c 29 23 8d df 00 0c 29 95 8a 50 08 00 45 00  ..)#.... )...P..E.
0010  00 98 7f 34 40 00 3f 32 1c ad c0 a8 0a 01 c0 a8  ...4@.?? .....
0020  14 01 71 79 cf 80 00 00 00 08 71 5b 49 08 5c c9  ..qy.... ..q[I.\.
0030  cc 38 ae 22 55 bc c3 bc a3 33 1e 98 d3 ab 67 af  .8."U... .3....g.
    
```

STEP: you can use setkey -D in both VMs to view the Security Association DB as below:-

```
root@vm2:~/Downloads
File Edit View Search Terminal Help
[root@vm2 Downloads]# setkey -D
192.168.10.1 192.168.20.1
    esp mode=tunnel spi=3752158016(0xdfa56340) reqid=16401(0x00004011
)
    E: aes-cbc 5ec6772e d4fd2dc1 20de531f 9f687dc5
    A: hmac-sha1 9a6984f8 bblfe028 5b6d59cf 2c5b1061 f53e9de3
    seq=0x00000000 replay=32 flags=0x00000000 state=mature
    created: Mar 28 00:52:22 2016    current: Mar 28 00:52:39 2016
    diff: 17(s)    hard: 0(s)    soft: 0(s)
    last:
    current: 0(bytes)    hard: 0(bytes)    soft: 0(bytes)
    allocated: 0    hard: 0    soft: 0
    sadb_seq=1 pid=8721 refcnt=0
192.168.20.1 192.168.10.1
    esp mode=tunnel spi=1696336313(0x651c09b9) reqid=16401(0x00004011
)
    E: aes-cbc 52a767f6 49f6d115 6fcd4f55 8a58cda4
    A: hmac-sha1 7deace3e b11e8342 0479448a e96cadb0 40c59197
    seq=0x00000000 replay=32 flags=0x00000000 state=mature
    created: Mar 28 00:52:22 2016    current: Mar 28 00:52:39 2016
    diff: 17(s)    hard: 0(s)    soft: 0(s)
    last:
    current: 0(bytes)    hard: 0(bytes)    soft: 0(bytes)
    allocated: 0    hard: 0    soft: 0
```

Links:-

IP SEC Tools:-

<http://ipsec-tools.sourceforge.net/>

How to use IPSEC Tools:-

<http://www.mad-hacking.net/documentation/linux/networking/ipsec/installation.xml>

libreswan:-

<https://libreswan.org/>

Oakley:-

<https://tools.ietf.org/html/rfc2412>