IPSec VPN

Task 01: Learn about VPN, and what is IKEv2

A VPN (Virtual Private Network) is a technology that creates a secure and private connection over a public network, such as the Internet. The VPN encrypts the data traffic and hides the online identity, making it difficult for third parties, such as hackers, ISPs (Internet Service Providers), or government agencies, to track user activities or steal user data.

How a VPN Works:

1. Encryption: When a VPN is used, the data is encrypted before it leaves the sending device. This means that even if someone intercepts the network traffic (MIM), the adversary won't be able to read or access the information.

2. VPN Server: The encrypted data is sent to a VPN server located in a different geographical location (which you can choose). The server decrypts the data and sends it to the destination website or service. The response from the website is then encrypted again and sent back to the user.

3. Hiding the IP Address: By connecting to a VPN server, the user's actual IP address (which reveals the user's location) is masked. Instead, websites see the IP address of the VPN server, making it appear as if the user is browsing from that server's location.

Why Use a VPN?

1. Privacy and Anonymity: A VPN hides a user's online activities from your ISP, government agencies, or anyone else who might be monitoring the user's connection. This can help protect browsing history, location, and personal data.

2. Security: VPNs encrypt the network connection, which is especially useful when using public Wi-Fi networks, such as those in coffee shops or airports. Encryption prevents hackers from intercepting sensitive information like passwords or credit card details.

3. Bypassing Censorship and Geo-Restrictions: VPNs allow you to access content that may be restricted or blocked in a location, such as certain websites, streaming services, or social media platforms. For example, one can watch region-specific content on Netflix or bypass internet censorship in some countries.

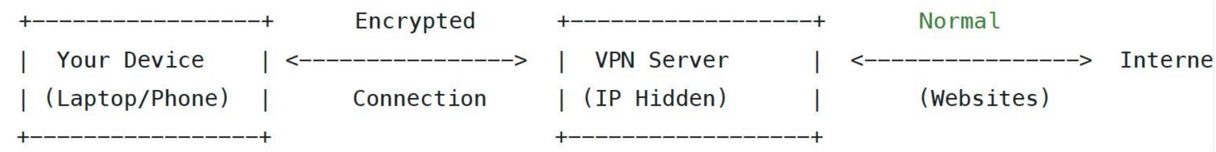
4. Secure Remote Access: Many businesses use VPNs to give employees secure access to the company's internal network when working remotely. This protects sensitive business information from being exposed.

5. Prevent Bandwidth Throttling: ISPs sometimes throttle (slow down) your internet speed based on user activities, such as streaming or downloading large files. Using a VPN can hide user activity from the ISP, potentially preventing throttling.

Limitations of VPNs:

- Reduced Speed: Encryption and rerouting data can slow down the network connection. - VPN Restrictions: Some websites and services actively block VPN traffic.

In summary, a VPN is a powerful tool for protecting online privacy, securing network traffic, and bypassing internet restrictions.



```
+-----------------+      Encrypted      +------------------+       Normal
|  Your Device    | <----------------->  |  VPN Server      |  <---------------->  Interne
| (Laptop/Phone)  |      Connection     | (IP Hidden)      |       (Websites)
+-----------------+                     +------------------+
```

IKEv2 (Internet Key Exchange version 2) is a protocol that helps establish and manage secure communications in a virtual private network (VPN). It is part of the IPsec (Internet Protocol Security) suite, which provides data encryption, authentication, and protection for secure communications over IP networks like the Internet.

Key Features of IKEv2:

1. Secure Key Exchange: IKEv2 is responsible for securely negotiating and exchanging keys between two parties (e.g., a client and a VPN server).

2. Efficient Connection Handling: It supports efficient handling of VPN connections, making it especially useful for devices that move between different networks, such as laptops and mobile phones.

3. Mobility and Multihoming: IKEv2 supports the MOBIKE (Mobility and Multihoming) protocol, which allows the VPN connection to maintain stability even when the user's IP address changes.

4. Strong Security: It uses strong encryption and authentication methods, such as AES (Advanced Encryption Standard) and digital certificates, to ensure secure communication.

5. Performance: IKEv2 is known for being more efficient and faster than its predecessor, IKEv1, with reduced latency and improved performance.

How IKEv2 Works:

IKEv2 operates in two main modes:

1. Mode 1: The VPN client and server authenticate each other and establish a secure communication channel. They negotiate cryptographic algorithms and generate shared secret keys.

2. Mode 2: The secure channel is used to establish one or more IPsec security associations (SAs), which are used to encrypt and decrypt the data transferred between the client and the server.

Benefits of IKEv2:

- Reliability: Good for maintaining long-lived connections even with changes in network connectivity.

- Compatibility: Works well with modern networks and is supported on many platforms, including Windows, macOS, iOS, and Android.

- Security: Offers robust security mechanisms, making it suitable for protecting sensitive data.

Overall, IKEv2 is a popular and secure choice for establishing VPN connections, especially for mobile and roaming users.

Task 02: Setup IKEv2 VPN Server

Carry out the following tasks on your VPN server.

```
apt-get update -y apt-get
upgrade -y
```

Once your VPN server is updated, edit the `/etc/sysctl.conf` file and enable the packet forwarding:

`sudo vim  /etc/sysctl.conf` (you can use nano or gedit in place of vim)

make the following changes in the `/etc/sysctl.conf`

```
net.ipv4.ip_forward=1
net.ipv4.conf.all.accept_redirects = 0 net.ipv4.conf.all.send_redirects
= 0
```

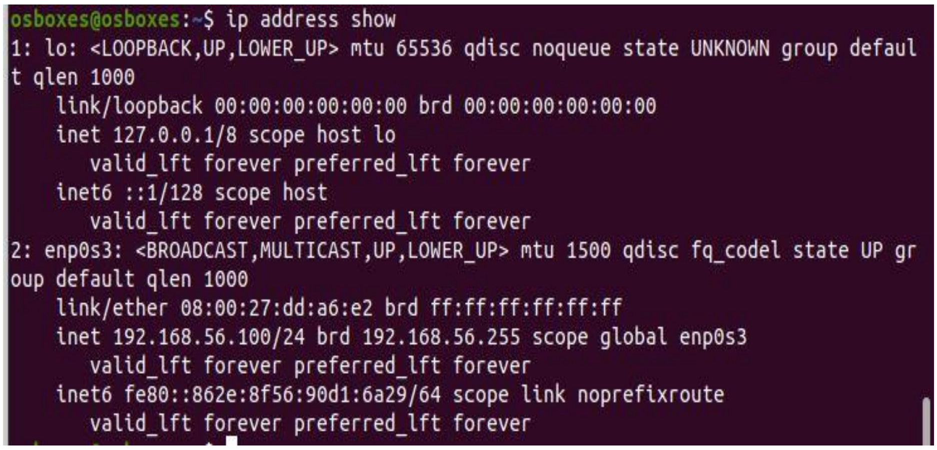
Save and close the file then run the following command to apply the configuration:

```
sudo sysctl -f
```

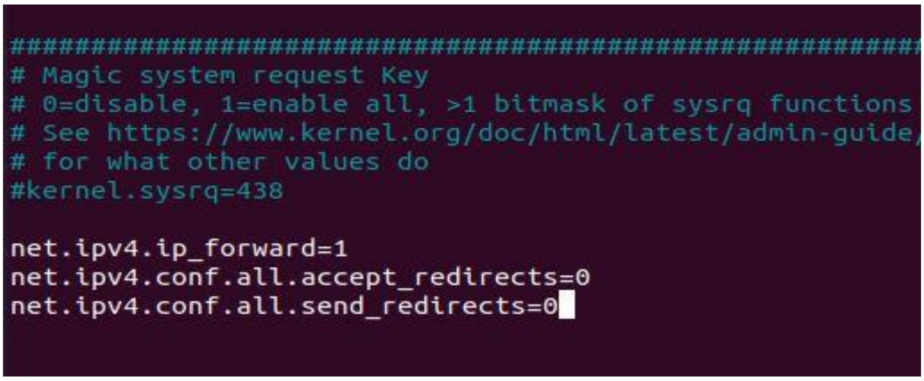
Checkpoint 01: Provide screenshot of edited /etc/sysctl.conf file.

Date:	

Evidence/Snapshot:
(Checkpoint)

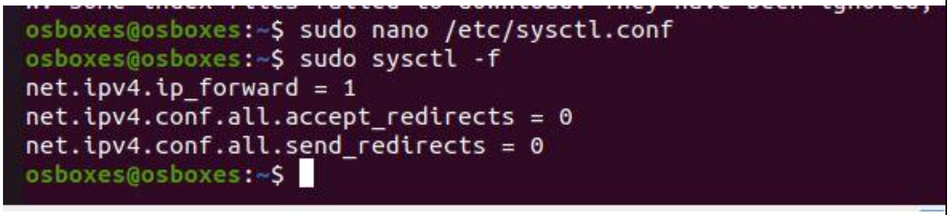


```
osboxes@osboxes:~$ ip address show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 08:00:27:dd:a6:e2 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.100/24 brd 192.168.56.255 scope global enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::862e:8f56:90d1:6a29/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```



```
###############################################################
# Magic system request Key
# 0=disable, 1=enable all, >1 bitmask of sysrq functions
# See https://www.kernel.org/doc/html/latest/admin-guide/
# for what other values do
#kernel.sysrq=438

net.ipv4.ip_forward=1
net.ipv4.conf.all.accept_redirects=0
net.ipv4.conf.all.send_redirects=0
```



```
osboxes@osboxes:~$ sudo nano /etc/sysctl.conf
osboxes@osboxes:~$ sudo sysctl -f
net.ipv4.ip_forward = 1
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.send_redirects = 0
osboxes@osboxes:~$
```

Description/Detail:
Static IP Address is Set
Required variables are given the given values

Note: Copy/Paste this table to provide multiple evidence or snapshots.

Task 02: Installing strongSwan, PKI packages and setting up the Certificate Authority (CA)

First, install the strongSwan and public key infrastructure (PKI) components on the VPN server. Install them by running the following command:

```
sudo apt install strongswan strongswan-pki libcharon-extra-plugins
libcharon-extauth-plugins libstrongswan-extra-plugins libtss2-
tctitabrmd-dev
```

Once all the packages are installed, we can create a VPN certificate.

A certificate for the IKEv2 server must be created to identify itself to the clients. The **strongswan-pki** provides a **PKI** utility that helps to create a CA and certificates.

First, set up the directories to save the CA and certificates.

```
sudo mkdir -p /root/pki/{cacerts,certs,private}
```

Next, generate a root key to sign the root certificate authority with the following command:

```
sudo pki --gen --type rsa --size 4096 --outform pem >
/root/pki/private/ca-key.pem
```

Use the root key and create a root certificate authority using the following command:

```
sudo pki --self --ca --lifetime 3650 --in /root/pki/private/ca-key.pem
--type rsa --dn "CN=VPN root CA" --outform pem >
/root/pki/cacerts/cacert.pem
```

Next, create a certificate and key for the VPN server. This certificate will be used to verify the server’s authenticity using the CA certificate.

Create a private key for the server using the following command:

```
sudo pki --gen --type rsa --size 4096 --outform pem >
/root/pki/private/server-key.pem
```

Next, create and sign the VPN server certificate using the CA that you created earlier:

```
sudo pki --pub --in /root/pki/private/server-key.pem --type rsa | pki -
issue --lifetime 1825 --cacert /root/pki/cacerts/ca-cert.pem --cakey
/root/pki/private/ca-key.pem --dn "CN=45.58.41.152" --san 45.58.41.152
--flag serverAuth --flag ikeIntermediate --outform pem >
/root/pki/certs/server-cert.pem
```

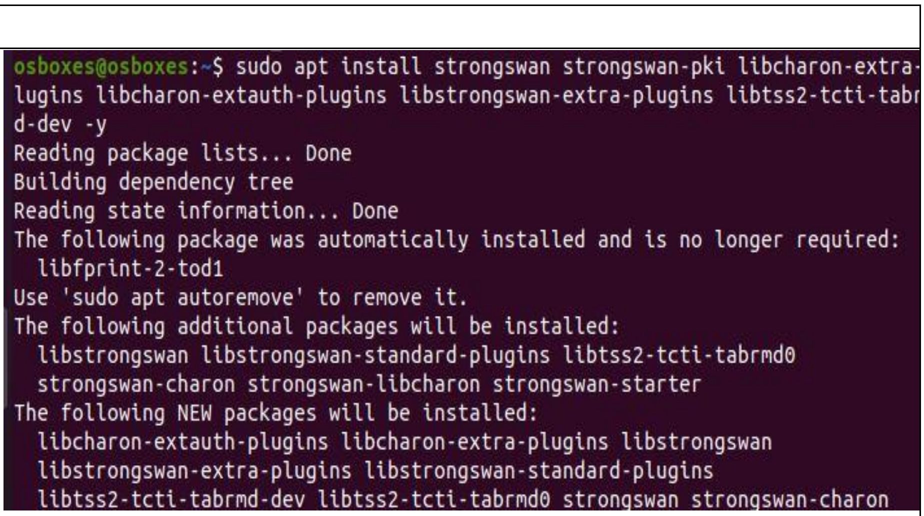
Now, copy all the certificates to the `/etc/ipsec.d` directory:

```
sudo cp -r /root/pki/* /etc/ipsec.d/
```

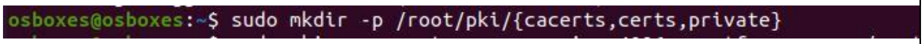
Checkpoint 02: Provide screenshots of the successful creation of certificates.

Date:

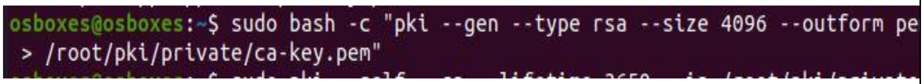
Evidence/Snapshot: *(Checkpoint)*



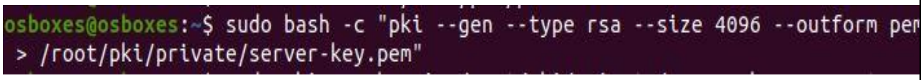
```
osboxes@osboxes:~$ sudo apt install strongswan strongswan-pki libcharon-extra-
lugins libcharon-extauth-plugins libstrongswan-extra-plugins libtss2-tcti-tabr
d-dev -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libfprint-2-tod1
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libstrongswan libstrongswan-standard-plugins libtss2-tcti-tabrmd0
  strongswan-charon strongswan-libcharon strongswan-starter
The following NEW packages will be installed:
  libcharon-extauth-plugins libcharon-extra-plugins libstrongswan
  libstrongswan-extra-plugins libstrongswan-standard-plugins
  libtss2-tcti-tabrmd-dev libtss2-tcti-tabrmd0 strongswan strongswan-charon
```



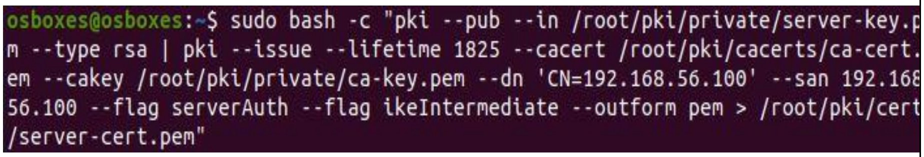
```
osboxes@osboxes:~$ sudo mkdir -p /root/pki/{cacerts,certs,private}
```



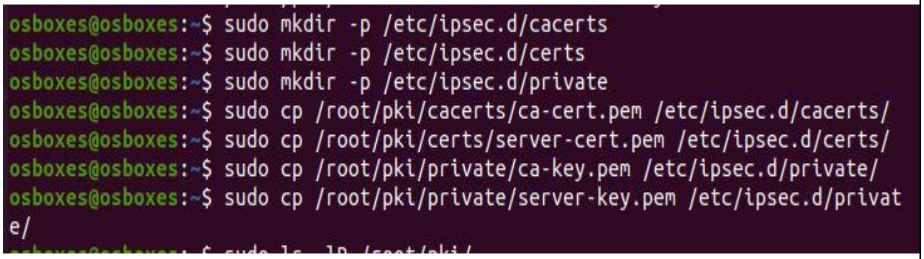
```
osboxes@osboxes:~$ sudo bash -c "pki --gen --type rsa --size 4096 --outform pe
 > /root/pki/private/ca-key.pem"
```



```
osboxes@osboxes:~$ sudo bash -c "pki --gen --type rsa --size 4096 --outform pem
 > /root/pki/private/server-key.pem"
```



```
osboxes@osboxes:~$ sudo bash -c "pki --pub --in /root/pki/private/server-key.p
m --type rsa | pki --issue --lifetime 1825 --cacert /root/pki/cacerts/ca-cert.
em --cakey /root/pki/private/ca-key.pem --dn 'CN=192.168.56.100' --san 192.168
56.100 --flag serverAuth --flag ikeIntermediate --outform pem > /root/pki/cert
/server-cert.pem"
```



```
osboxes@osboxes:~$ sudo mkdir -p /etc/ipsec.d/cacerts
osboxes@osboxes:~$ sudo mkdir -p /etc/ipsec.d/certs
osboxes@osboxes:~$ sudo mkdir -p /etc/ipsec.d/private
osboxes@osboxes:~$ sudo cp /root/pki/cacerts/ca-cert.pem /etc/ipsec.d/cacerts/
osboxes@osboxes:~$ sudo cp /root/pki/certs/server-cert.pem /etc/ipsec.d/certs/
osboxes@osboxes:~$ sudo cp /root/pki/private/ca-key.pem /etc/ipsec.d/private/
osboxes@osboxes:~$ sudo cp /root/pki/private/server-key.pem /etc/ipsec.d/privat
e/
```

Have you used another tool in the past labs to set up a CA? describe briefly what you have learned about PKI command?

The `pki` commands in strongSwan help you **create, manage, and verify certificates** used for securing communication in a VPN setup (IPsec). These certificates are part of the **Public Key Infrastructure (PKI)** system, where each participant (server or client) has a private and public key.

Description/Detail:

All the certificates are made successfully

And then copied too, but the copy command was not working in one go. So divied it in parts and then applied

Note: Copy/Paste this table to provide multiple evidence or snapshots.

At this point, all certificates and CA are required by strongSwan to secure communications between the client and the server are created. Next, proceed to configure the strongSwan VPN server.

Task 03: Configure strongSwan VPN Server

In this task, you will learn to configure and set up IKEv2 VPN Server. It is recommended to preserve the default configuration file and create a new configuration file. To make a backup copy of the strongSwan default configuration file, run the following command:

```
sudo mv /etc/ipsec.conf /etc/ipsec.conf.bak
```

Next, create a new configuration file using the following command:

`sudo vim /etc/ipsec.conf` (you can use nano or gedit in place of vim)

and add the following configurations:

```
config setup
charondebug="ike 1, knl 1, cfg 0" uniqueids=no
 conn ikev2-vpn auto=add
compress=no type=tunnel
keyexchange=ikev2
fragmentation=yes
forceencaps=yes
dpdaction=clear
dpddelay=300s rekey=no
left=%any
leftid=192.168.56.100
leftcert=server-cert.pem
leftsendcert=always
leftsubnet=0.0.0.0/0
right=%any rightid=%any
rightauth=eap-mschapv2
rightsourceip=10.10.10.0
/24
rightdns=8.8.8.8,8.8.4.4
rightsendcert=never
eap_identity=%identity
ike=chacha20poly1305-sha512-curve25519-prfsha512,aes256gcm16-
sha384prfsha384-ecp384,aes256-sha1-modp1024,aes128-sha1-modp1024,3des-
sha1modp1024!
esp=chacha20poly1305-sha512,aes256gcm16-ecp384,aes256-
sha256,aes256sha1,3des-sha1!
```

save and close the file when you are finished.

A brief explanation of each option is shown below:

```
left=%any - The %any means the server will use any network interface to
receive incoming connections.
 leftid=192.168.56.100 - Specify the IP address of the VPN
server.
 leftcert=server-cert.pem - Specify the name of the public
certificate.

leftsendcert=always - The always means that any remote clients will
receive a copy of the server's public certificate.

leftsubnet=0.0.0.0/0 - It specifies the entire set of IPv4 addresses

rightauth=eap-mschapv2 - Define the authentication method used by the
client to authenticate the server.

rightsourceip=10.10.10.0/24 - This will tell the server to assign
private IP to clients from the 10.10.10.0/24 network.
 rightdns=8.8.8.8,8.8.4.4 - It specifies Google's DNS IP
address.
```

Next, configure the authentication mechanism for strongSwan VPN. Edit the `ipsec.secrets` file and define the name of the private key file and define the user that is allowed to connect to the VPN server.

`sudo vim /etc/ipsec.secrets` (you can use nano or gedit in place of vim)

add the following lines:

```
: RSA "server-key.pem"
vpnusername : EAP "SomeSecurePassword"
```

Save and close the file and then restart the strongSwan service with the following command:

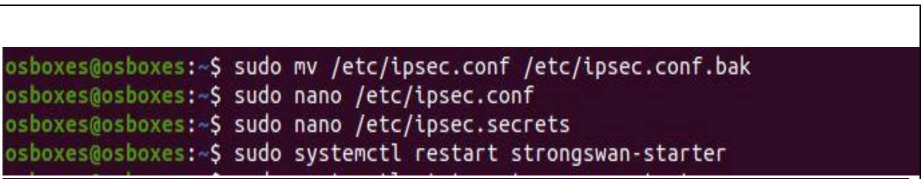
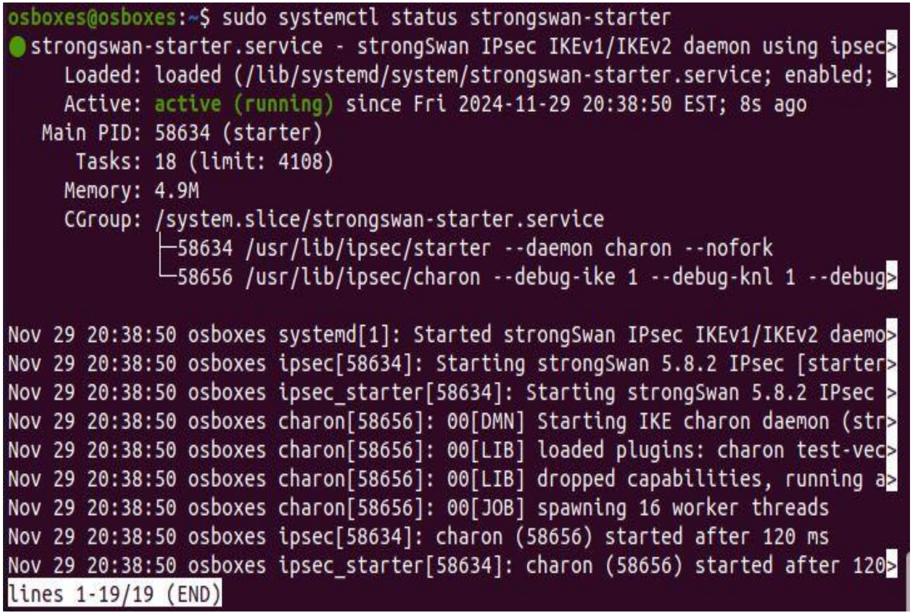
```
sudo systemctl restart strongswan-starter
```

Check the status of the strongSwan VPN service for any configuration error using the following command:

```
sudo systemctl status strongswan-starter
```

Checkpoint 03: Provide screenshots of the successful running status of strongSwoan.

Date:	

<table>
<tr><td><b><i>Evidence/Snapshot:</i></b> <i>(Checkpoint)</i></td><td>

<pre>
osboxes@osboxes:~$ sudo mv /etc/ipsec.conf /etc/ipsec.conf.bak
osboxes@osboxes:~$ sudo nano /etc/ipsec.conf
osboxes@osboxes:~$ sudo nano /etc/ipsec.secrets
osboxes@osboxes:~$ sudo systemctl restart strongswan-starter
</pre>

<pre>
osboxes@osboxes:~$ sudo systemctl status strongswan-starter
● strongswan-starter.service - strongSwan IPsec IKEv1/IKEv2 daemon using ipsec>
     Loaded: loaded (/lib/systemd/system/strongswan-starter.service; enabled; >
     Active: active (running) since Fri 2024-11-29 20:38:50 EST; 8s ago
   Main PID: 58634 (starter)
      Tasks: 18 (limit: 4108)
     Memory: 4.9M
     CGroup: /system.slice/strongswan-starter.service
             ├─58634 /usr/lib/ipsec/starter --daemon charon --nofork
             └─58656 /usr/lib/ipsec/charon --debug-ike 1 --debug-knl 1 --debug>

Nov 29 20:38:50 osboxes systemd[1]: Started strongSwan IPsec IKEv1/IKEv2 daemo>
Nov 29 20:38:50 osboxes ipsec[58634]: Starting strongSwan 5.8.2 IPsec [starter>
Nov 29 20:38:50 osboxes ipsec_starter[58634]: Starting strongSwan 5.8.2 IPsec >
Nov 29 20:38:50 osboxes charon[58656]: 00[DMN] Starting IKE charon daemon (str>
Nov 29 20:38:50 osboxes charon[58656]: 00[LIB] loaded plugins: charon test-vec>
Nov 29 20:38:50 osboxes charon[58656]: 00[LIB] dropped capabilities, running a>
Nov 29 20:38:50 osboxes charon[58656]: 00[JOB] spawning 16 worker threads
Nov 29 20:38:50 osboxes ipsec[58634]: charon (58656) started after 120 ms
Nov 29 20:38:50 osboxes ipsec_starter[58634]: charon (58656) started after 120>
lines 1-19/19 (END)
</pre>
</td></tr>
<tr><td><b><i>Description/Detail:</i></b></td><td><i>strongmanSwan Successfully RUNNING</i></td></tr>
<tr><td colspan="2"><i>Note: Copy/Paste this table to provide multiple evidence or snapshots.</i></td></tr>
</table>

At this point, strongSwan VPN server is installed, configured and ready for use by a client.

Task 04: Install and Configure strongSwan VPN Client and connect.

In this task, you will learn to install, configure strongSwan client package and connect it to the strongSwan VPN server.

Install the strongSwan VPN client package using the following command on the client machine:

```
sudo apt-get install strongswan libcharon-extra-plugins -y
```

Copy the CA certificate file from the server machine to the client machine:

```
scp root@192.168.56.100:/etc/ipsec.d/cacerts/ca-cert.pem
/etc/ipsec.d/cacerts
```

Next, edit the `ipsec.secrets` file and provide your username and password which you have defined on the server machine.

```
sudo vim /etc/ipsec.secrets
```

Add the following to the file:

```
vpnusername : EAP "SomeSecurePassword"
```

Save and close the file, and edit the strongSwan configuration file with the following command:

```
sudo vim /etc/ipsec.conf
```

Add the following to the file:

```
conn ipsec-ikev2-vpn-client
auto=start right=192.168.56.100
rightid=192.168.56.100
rightsubnet=0.0.0.0/0
rightauth=pubkey
leftsourceip=%config
leftid=vpnusername
leftauth=eap-mschapv2
eap_identity=%identity
```

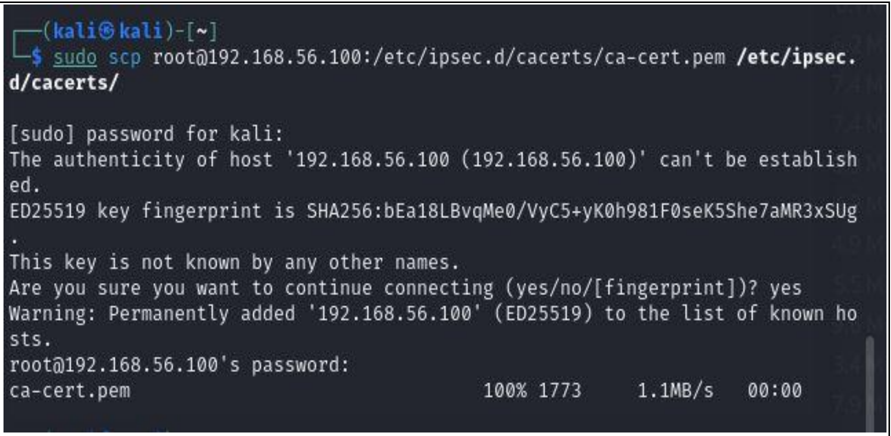
Save and close the file and restart the strongSwan service.

```
sudo systemctl restart strongswan-starter
```

Checkpoint 04: Provide screenshots of the successful running status of strongSwoan on the Client.

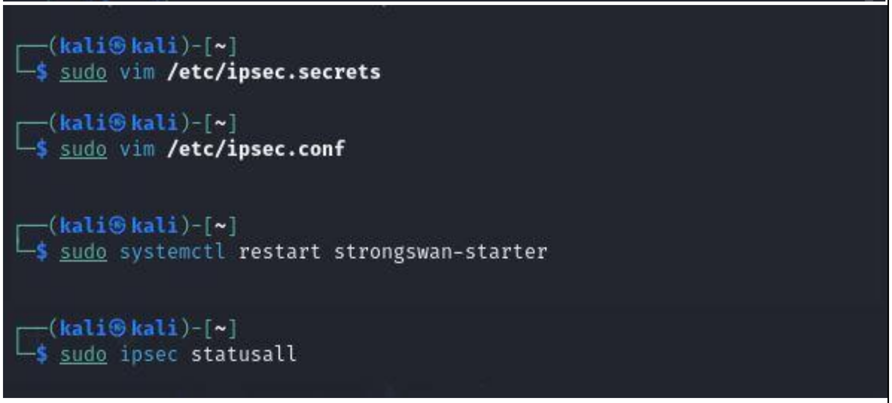
Date:	

Evidence/Snapshot: *(Checkpoint)*



```
┌──(kali㉿kali)-[~]
└─$ sudo scp root@192.168.56.100:/etc/ipsec.d/cacerts/ca-cert.pem /etc/ipsec.
d/cacerts/

[sudo] password for kali:
The authenticity of host '192.168.56.100 (192.168.56.100)' can't be establish
ed.
ED25519 key fingerprint is SHA256:bEa18LBvqMe0/VyC5+yK0h981F0seK5She7aMR3xSUg
.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.100' (ED25519) to the list of known ho
sts.
root@192.168.56.100's password:
ca-cert.pem                                   100% 1773     1.1MB/s   00:00
```



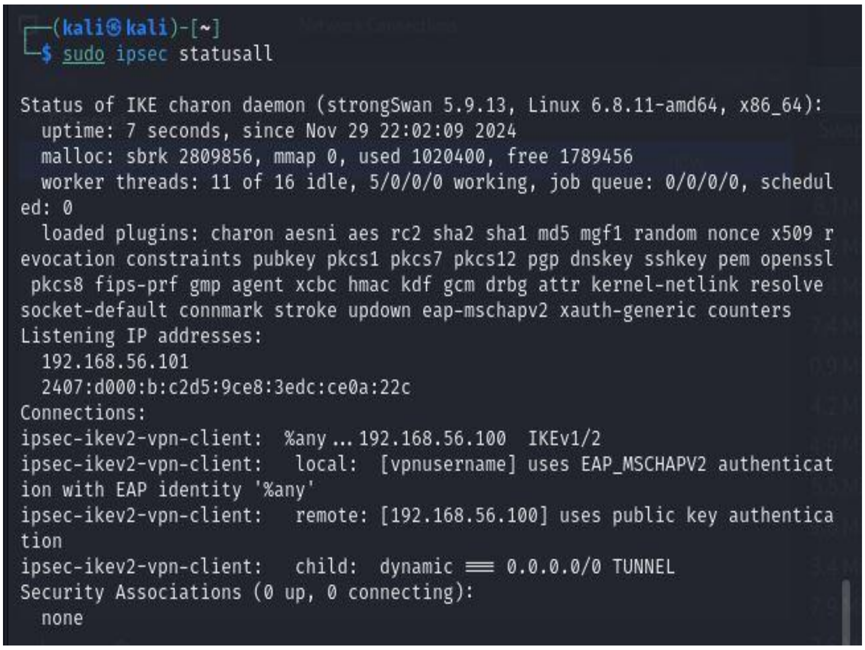
```
┌──(kali㉿kali)-[~]
└─$ sudo vim /etc/ipsec.secrets

┌──(kali㉿kali)-[~]
└─$ sudo vim /etc/ipsec.conf


┌──(kali㉿kali)-[~]
└─$ sudo systemctl restart strongswan-starter


┌──(kali㉿kali)-[~]
└─$ sudo ipsec statusall
```

use `ipsec statusall` command to see the status



```
┌──(kali㉿kali)-[~]
└─$ sudo ipsec statusall

Status of IKE charon daemon (strongSwan 5.9.13, Linux 6.8.11-amd64, x86_64):
  uptime: 7 seconds, since Nov 29 22:02:09 2024
  malloc: sbrk 2809856, mmap 0, used 1020400, free 1789456
  worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, schedul
ed: 0
  loaded plugins: charon aesni aes rc2 sha2 sha1 md5 mgf1 random nonce x509 r
evocation constraints pubkey pkcs1 pkcs7 pkcs12 pgp dnskey sshkey pem openssl
 pkcs8 fips-prf gmp agent xcbc hmac kdf gcm drbg attr kernel-netlink resolve
socket-default connmark stroke updown eap-mschapv2 xauth-generic counters
Listening IP addresses:
  192.168.56.101
  2407:d000:b:c2d5:9ce8:3edc:ce0a:22c
Connections:
ipsec-ikev2-vpn-client:  %any...192.168.56.100  IKEv1/2
ipsec-ikev2-vpn-client:   local:  [vpnusername] uses EAP_MSCHAPV2 authenticat
ion with EAP identity '%any'
ipsec-ikev2-vpn-client:   remote: [192.168.56.100] uses public key authentica
tion
ipsec-ikev2-vpn-client:   child:  dynamic === 0.0.0.0/0 TUNNEL
Security Associations (0 up, 0 connecting):
  none
```

Description/Detail:

Note: Copy/Paste this table to provide multiple evidence or snapshots.