StrongSwan

Installation

We'll install StrongSwan (https://www.strongswan.org) , an open-source IPSec daemon which we'll configure as our VPN server. We'll also install the public key infrastructure component so that we can create a certificate authority to provide credentials for our infrastructure. So, in Ubuntu you should execute these commands...

```
sudo apt update
sudo apt install strongswan strongswan-pki
```

...and in Fedora just this one:

```
sudo dnf install strongswan
```

PKI infraestructure

An IKEv2 server requires a certificate to identify itself to clients. To help us create the certificate required, the strongswan-pki package comes with a utility to generate a certificate authority and server certificates. To begin, let's create a few directories to store all the assets we'll be working on. The directory structure matches some of the directories in /etc/ipsec.d, where we will eventually move all of the items we create. We'll lock down the permissions so that our private files can't be seen by other users:

```
mkdir -p ~/pki/{cacerts,certs,private}
chmod 700 ~/pki
```

Now that we have a directory structure to store everything, we can generate a root key. This will be a 4096-bit RSA key that will be used to sign our root certificate authority. Execute these commands to generate the key:

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

Now that we have a key, we can move on to creating our root certificate authority, using the key to sign the root certificate:

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

You can change the distinguished name (DN) values to something else to if you would like. The common name here is just the indicator, so it doesn't have to match anything in your infrastructure. Now that we've got our root certificate authority up and running, we can create a certificate that the VPN server will use. This certificate will allow the client to verify the server's authenticity using the CA certificate we just generated. First, create a private key for the VPN server with the following command:

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

Now, create and sign the VPN server certificate with the certificate authority's key you created in the previous step. Execute the following command, but change the Common Name (CN) and the Subject Alternate Name (SAN) field to your VPN server's DNS name or IP address:

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

Now that we've generated all of the TLS/SSL files StrongSwan needs, we can move the files into place in the /etc/ipsec.d directory by typing:

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

Server configuration

StrongSwan has a default configuration file with some examples, but we will have to do most of the configuration ourselves. Let's back up the file for reference before starting from scratch:

```
sudo mv /etc/ipsec.conf{,.original}
```

Create and open a new blank configuration file by typing:

```
sudo nano /etc/ipsec.conf
```

First, we'll tell StrongSwan to log daemon statuses for debugging and allow duplicate connections. Add these lines to the file:

```
config setup
    charondebug="ike 1, knl 1, cfg 0"
    uniqueids=no
```

Then, we'll create a configuration section for our VPN. We'll also tell StrongSwan to create IKEv2 VPN Tunnels and to automatically load this configuration section when it starts up. Append the following lines to the file:

```
. . .
conn ikev2-vpn
    auto=add
    compress=no
    type=tunnel
    keyexchange=ikev2
    fragmentation=yes
    forceencaps=yes
```

We'll also configure dead-peer detection to clear any "dangling" connections in case the client unexpectedly disconnects. Add these lines:

```
. . .
conn ikev2-vpn
    . . .
    dpdaction=clear
    dpddelay=300s
    rekey=no
```

Then, we'll configure the server (left) side IPSec parameters. Add this to the file:

```
. . .
conn ikev2-vpn
    . . .
    left=%any
    leftid=@server_domain_or_IP
    leftcert=server-cert.pem
    leftsendcert=always
    leftsubnet=0.0.0.0/0
```

> NOTA: When configuring the server ID (leftid), only include the @ character if your VPN server will be identified by a domain name: leftid=@vpn.example.com If the server will be identified by its IP address, just put the IP address in: *leftid=203.0.113.7*

Next, we can configure the client (right) side IPSec parameters, like the private IP address ranges and DNS servers to use:

```
. . .
conn ikev2-vpn
    . . .
    right=%any
    rightid=%any
```

rightauth=eap-mschapv2
rightsourceip=10.10.10.0/24
rightdns=8.8.8.8,8.8.4.4
rightsendcert=never

Finally, we'll tell StrongSwan to ask the client for user credentials when they connect:

. . .
conn ikev2-vpn
. . .
eap_identity=%identity

Now that we've configured the VPN parameters, let's move on to creating an account so our users can connect to the server.

Credentials configuration

Our VPN server is now configured to accept client connections, but we don't have any credentials configured yet. We'll need to configure a couple things in a special configuration file called **ipsec.secrets**:

*We need to tell StrongSwan where to find the private key for our server certificate, so the server will be able to authenticate to clients.
*We also need to set up a list of users that will be allowed to connect to the VPN.

Let's open the secrets file for editing:

sudo nano /etc/ipsec.secrets

First, we'll tell StrongSwan where to find our private key:

: RSA "server-key.pem"

Then, we'll define the user credentials. You can make up any username or password combination that you like:

your_username : EAP "your_password"

Save and close the file. Now that we've finished working with the VPN parameters, we'll restart the VPN service so that our configuration is applied:

sudo systemctl restart strongswan

Firewall configuration

You should open the standard IPSec ports: 500/UDP and 4500/UDP. Moreover, you'll need to add a few low-level policies for routing and forwarding IPSec packets. Before we do, we need to find which network interface on our server is used for internet access. We can find that by querying for the interface associated with the default route:

ip route | grep default

Your public interface should follow the word "dev" (let's suppose it's "eth0"). So you should write these firewall rules:

iptables -t nat -A POSTROUTING -s 10.10.10.0/24 -o eth0 -m policy --pol ipsec --dir out -j ACCEPT
iptables -t nat -A POSTROUTING -s 10.10.10.0/24 -o eth0 -j MASQUERADE
iptables -t mangle -A FORWARD --match policy --pol ipsec --dir in -s 10.10.10.0/24 -o eth0 -p tcp -m tcp --tcp-flags SYN,RST SYN -m tcpmss --mss 1361:1536 -j TCPMSS --set-mss 1360

The *nat lines create rules so that the firewall can correctly route and manipulate traffic between the VPN clients and the internet. The *mangle line adjusts the maximum packet segment size to prevent potential issues with certain VPN clients.

and

iptables -t filter -A ufw-before-forward --match policy --pol ipsec --dir in --proto esp -s 10.10.10.0/24 -j ACCEPT
iptables -t filter -A ufw-before-forward --match policy --pol ipsec --dir out --proto esp -d 10.10.10.0/24 -j ACCEPT

These lines tell the firewall to forward ESP (Encapsulating Security Payload) traffic so the VPN clients will be able to connect. ESP provides additional security for our VPN packets as they're traversing untrusted networks.

Finally, we'll need to configure a few things in sysctl.conf:

*First, we'll enable IPv4 packet forwarding (net/ipv4/ip_forward=1)
*We'll disable Path MTU discovery to prevent packet fragmentation problems (net/ipv4/ip_no_pmtu_disc=1)
*We also won't send ICMP redirects (net/ipv4/conf/all/send_redirects=0) nor accept ICMP redirects (net/ipv4/conf/all/accept_redirects=0) as we aren't routers. This is to prevent man-in-the-middle attacks

Client configuration

Copy the CA certificate you created and install it on your client device(s) that will connect to the VPN. The easiest way to do this is to log into your server and save the contents of the certificate file, including the -----BEGIN CERTIFICATE----- and -----END CERTIFICATE----- lines, to a file with a recognizable name, such as "ca-cert.pem". Ensure the file you create has the .pem extension. Alternatively, you could use SFTP to transfer the file to your computer. Once you have the ca-cert.pem file downloaded to your computer, you can set up the connection to the VPN.

To connect from an Ubuntu machine, you can set up and manage StrongSwan as a service or use a one-off command every time you wish to connect. Instructions are provided for both.

*Managing StrongSwan as a Service:

1. Update your local package cache: *sudo apt update*
2. Install StrongSwan and the related software: *sudo apt install strongswan libcharon-extra-plugins*
3. Copy the CA certificate to the /etc/ipsec.d/cacerts directory: *sudo cp /tmp/ca-cert.pem /etc/ipsec.d/cacerts*
4. Disable StrongSwan so that the VPN doesn't start automatically: *sudo systemctl disable --now strongswan*
5. Configure your VPN username and password in the /etc/ipsec.secrets file: your_username *: EAP "your_password"*
6. Edit the /etc/ipsec.conf file to define your configuration.

```
config setup
conn ikev2-rw
  right=server_domain_or_IP
  # This should match the `leftid` value on your server's configuration
  rightid=server_domain_or_IP
  rightsubnet=0.0.0.0/0
  rightauth=pubkey
  leftsourceip=%config
  leftid=username
  leftauth=eap-mschapv2
  eap_identity=%identity
  auto=start
```

7. To connect to the VPN, type: *sudo systemctl start strongswan*
8. To disconnect again, type: *sudo systemctl stop strongswan*

*Using a Simple Client for One-Off Connections:

1. Update your local package cache: *sudo apt update*
2. Install charon-cmd and related software *sudo apt install charon-cmd libcharon-extra-plugins*
3. Move to the directory where you copied the CA certificate: *cd /path/to/ca-cert.pem*
4. Connect to the VPN server with charon-cmd using the server's CA certificate, the VPN server's IP address, and the username you configured: *sudo charon-cmd --cert ca-cert.pem --host vpn_domain_or_IP --identity your_username*
5. When prompted, provide the VPN user's password.
6. You should now be connected to the VPN. To disconnect, press **CTRL+C** and wait for the connection to close.

Podeu trobar més informació a https://www.ivpn.net/setup/gnu-linux-ipsec-with-ikev2.html