## Configure IPSEC VPN using StrongSwan on Ubuntu 18.04

### Install strongSwan on Ubuntu 18.04

Fortunately, strongSwan is available on the default Ubuntu 18.04 repositories and thus can simply be installed by running the command below;

apt install strongswan libcharon-extra-plugins

### Setup CA Using the strongSwan PKI Tool

In order for the VPN client to verify the authenticity of the VPN server, you need to generate the VPN server certificate and key. Before you can generate the server certificate and the key, you have to create a local CA for signing them. stronSwan provides a PKI utility that eases this process. However you need to install this utility by running the command below;

apt install strongswan-pki

Once the installation is done, proceed to create the CA. To begin with, generate a private key for self-signing the CA certificate.

ipsec pki --gen --size 4096 --type rsa --outform pem > vpn-ca.key.pem

Ensure that you accord this key the absolute privacy it deserves.

Generate the VPN server CA and self-sign with the key generated above.

ipsec pki --self --in vpn-ca.key.pem --type rsa --dn "CN=VPN Server root CA" --ca --lifetime 3650 --outform pem > vpn-ca.cert.pem

Next, generate a VPN server private key and issue a matching certificate using the CA created above.

ipsec pki --gen --size 4096 --type rsa --outform pem > vpn-server.key.pem

Once you have the server key in place, generate the server certificate by running the command below. Be sure to replace the DN and SAN accordingly.

ipsec pki --pub --in vpn-server.key.pem --type rsa \ |

ipsec pki --issue --lifetime 2750 \

--cacert vpn-ca.cert.pem \

--cakey vpn-ca.key.pem \

--dn "CN=vpnsvr.example.com" \

--san="vpnsvr.example.com" \

--flag serverAuth --flag ikeIntermediate --outform pem > vpn-server.cert.pem

### Install the certificates

Now that you got all certificates, you can install them by moving them to the respective IPSec certificates directories under /etc/ipsec.d.

mv vpn-ca.cert.pem /etc/ipsec.d/cacerts/

mv vpn-server.cert.pem /etc/ipsec.d/certs/

mv {vpn-ca.key.pem,vpn-server.key.pem} /etc/ipsec.d/private/

### Configure StrongSwan on Ubuntu 18.04

/etc/ipsec.conf configuration file specifies most configuration and control information for the strongSwan IPsec subsystem.It consists of three different section types:

* CONFIG SECTIONS (config setup)  
  – It defines general configuration parameters
* CONN SECTIONS (conn <name>)  
  – A conn section contains a connection specification, defining a network connection to be made using IPsec.
* CA SECTION (ca <name>)  
  – It defines a certification authority.

Before you can configure this file, make a backup of it.

cp /etc/ipsec.conf /etc/ipsec.conf.bak

vim /etc/ipsec.conf

Define the CONFIGURATION parameters;

config setup

charondebug="ike 2, knl 2, cfg 2, net 2, esp 2, dmn 2, mgr 2"

strictcrlpolicy=no

uniqueids=yes

cachecrls=no

* The charondebug = <debug list>  parameter defines the charon debug loggin where the debug list can be dmn, mgr, ike, chd, job, cfg, knl, net, asn, enc, lib, esp, tls, tnc, imc, imv, pts. The logging levels can one of -1, 0, 1, 2, 3, 4 (for silent, audit, control, controlmore, raw, private). By default, the level is set to 1 for all types. For a description of the debug lists, check the LOGGER CONFIGURATION section on strongswan.conf(5).
* strictcrlpolicy parameter defines if a fresh CRL must be available in order for the peer authentication based on RSA signatures to succeed.
* uniqueids defines whether a particular participant ID should be kept unique
* cachecrls defines whether to or not cache the certificate revocation lists (CRLs) fetched via HTTP or LDAP.

Define the CONNECTION parameters;

conn ipsec-ikev2-vpn

auto=add

compress=no

type=tunnel # defines the type of connection, tunnel.

keyexchange=ikev2

fragmentation=yes

forceencaps=yes

dpdaction=clear

dpddelay=300s

rekey=no

left=%any

leftid=@vpnsvr.example.com # if using IP, define it without the @ sign

leftcert=vpn-server.cert.pem # reads the VPN server cert in /etc/ipsec.d/certs

leftsendcert=always

leftsubnet=0.0.0.0/0

right=%any

rightid=%any

rightauth=eap-mschapv2

rightsourceip=10.10.7.0/24 # IP address Pool to be assigned to the clients

rightdns=8.8.8.8 # DNS to be assigned to clients

rightsendcert=never

eap\_identity=%identity # defines the identity the client uses to reply to an EAP Identity request.

To see a comprehensive description of the connection parameters and the values used in the above configuration, see man ipsec.conf.

Next, you need to configure client-server authentication credentials. The authentication credentials  are set in the /etc/ipsec.secrets configuration file. Thus open this file and define the RSA private keys for authentication. You can also setup the EAP user credentials by defining a random username and its password. Note the Spacing.

vim /etc/ipsec.secrets

# This file holds shared secrets or RSA private keys for authentication.

# RSA private key for this host, authenticating it to any other host

# which knows the public part.**: RSA vpn-server.key.pem #** VPN server key generated above

# <user id> : EAP <secret>**vpnsecure : EAP "P@sSw0Rd" #** Random

Save the configuration file and restart strongSwan for the changes to take effect.

systemctl restart strongswan

To verify that strongSwan has the private key in place, run the command below;

ipsec listcerts

List of X.509 End Entity Certificates

subject: "CN=vpnsvpnsvr.example.com"

issuer: "CN=VPN Server root CA"

validity: not before Feb 09 20:27:18 2019, ok

not after Aug 21 20:27:18 2026, ok (expires in 2749 days)

serial: 56:e5:08:a6:db:f6:6b:d0

altNames: vpnsvr.example.com

flags: serverAuth ikeIntermediate

authkeyId: 68:40:92:5b:53:c4:99:18:3e:7e:cb:6b:5b:32:d5:05:f7:de:88:74

subjkeyId: 09:2f:bd:61:bd:47:1b:c8:13:e0:2f:65:c0:9f:12:7b:0e:e8:c4:9b

pubkey: RSA 4096 bits, has private key

keyid: 75:72:19:89:62:97:27:55:a0:4f:68:be:6a:c9:14:98:04:87:be:a3

subjkey: 09:2f:bd:61:bd:47:1b:c8:13:e0:2f:65:c0:9f:12:7b:0e:e8:c4:9b

### Configure Firewall and Routing

Set UFW to allow and forward the VPN traffic. For IPsec to work through firewall, you need to open UDP ports 500 and 4500.

ufw allow 500/udp # Allows Internet Security Association and Key Management Protocol (ISAKMP) traffic to be forwarded

ufw allow 4500/udp # Allows handling of IPsec between natted devices

Find the default routing interface;

ip route show default

default via 10.0.2.2 dev enp0s3

So the interface, enp0s3 is the default routing interface.

Next, edit the **/etc/ufw/before.rules** such that your configuration looks like below. **Replace the IP pool and the default route interface accordingly**. See the highlighted lines added immediately before and after the \*filter.

...

# Don't delete these required lines, otherwise there will be errors**\*nat-A POSTROUTING -s 10.10.7.0/24 -o enp0s3 -m policy --pol ipsec --dir out -j ACCEPT-A POSTROUTING -s 10.10.7.0/24 -o enp0s3 -j MASQUERADECOMMIT\*mangle-A FORWARD --match policy --pol ipsec --dir in -s 10.10.7.0/24 -o enp0s3 -p tcp -m tcp --tcp-flags SYN,RST SYN -m tcpmss --mss 1361:1536 -j TCPMSS --set-mss 1360COMMIT**

\*filter

:ufw-before-input - [0:0]

:ufw-before-output - [0:0]

:ufw-before-forward - [0:0]

:ufw-not-local - [0:0]

# End required lines**-A ufw-before-forward --match policy --pol ipsec --dir in --proto esp -s 10.10.7.0/24 -j ACCEPT**

**-A ufw-before-forward --match policy --pol ipsec --dir out --proto esp -d 10.10.7.0/24 -j ACCEPT**

Disable and enable ufw for the changes to take effect.

ufw disable && ufw enable

Configure Kernel IP forwarding.

vim /etc/ufw/sysctl.conf

# Uncomment the next line to enable packet forwarding for IPv4

net/ipv4/ip\_forward=1

...

# Do not accept ICMP redirects (prevent MITM attacks)

net/ipv4/conf/all/accept\_redirects = 0

...

# Do not send ICMP redirects (we are not a router)

net/ipv4/conf/all/send\_redirects = 0

...

Disble Path MTU discovery to prevent packet fragmentation by adding the line below

net/ipv4/ip\_no\_pmtu\_disc=1

Restart StrongSwan and enable it to run on boot.

systemctl restart strongswan

systemctl enable strongswan

You can check the status.

systemctl status strongswan● strongswan.service - strongSwan IPsec IKEv1/IKEv2 daemon using ipsec.conf

Loaded: loaded (/lib/systemd/system/strongswan.service; enabled; vendor preset: enabled)

Active: active (running) since Sun 2019-02-10 10:15:44 UTC; 11s ago

Main PID: 9168 (starter)

Tasks: 18 (limit: 2325)

CGroup: /system.slice/strongswan.service

├─9168 /usr/lib/ipsec/starter --daemon charon --nofork

└─9193 /usr/lib/ipsec/charon --debug-ike 2 --debug-knl 2 --debug-cfg 2 --debug-net 2 --debug-esp 2 --debug-dmn 2 --debug-mgr 2

You can also check the status using the ipsec command.

ipsec statusall

Status of IKE charon daemon (strongSwan 5.6.2, Linux 4.15.0-45-generic, x86\_64):

uptime: 2 minutes, since Feb 10 10:15:44 2019

malloc: sbrk 1486848, mmap 0, used 501040, free 985808

worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled: 0

loaded plugins: charon aesni aes rc2 sha2 sha1 md4 md5 mgf1 random nonce x509 revocation constraints pubkey pkcs1 pkcs7 pkcs8 pkcs12 pgp dnskey sshkey pem openssl fips-prf gmp agent xcbc hmac gcm attr kernel-netlink resolve socket-default connmark stroke updown eap-mschapv2 xauth-generic counters

Virtual IP pools (size/online/offline):

10.10.7.0/24: 254/0/0

Listening IP addresses:

192.168.43.85

Connections:

ipsec-ikev2-vpn: %any...%any IKEv2, dpddelay=300s

ipsec-ikev2-vpn: local: [vpnsvr.example.com] uses public key authentication

ipsec-ikev2-vpn: cert: "CN=vpnsvr.example.com"

ipsec-ikev2-vpn: remote: uses EAP\_MSCHAPV2 authentication with EAP identity '%any'

ipsec-ikev2-vpn: child: 0.0.0.0/0 === dynamic TUNNEL, dpdaction=clear

Security Associations (0 up, 0 connecting):

none

### Testing VPN Connection

Now that we have configured  IPSEC VPN using strongSwan on Ubuntu 18.04, let us test if the remote clients can connect to it. In this guide, we are testing the connection from an Ubuntu 18.04 client.

Fire up an Ubuntu 18.04 client and install the following packages.

apt update

apt install strongswan libcharon-extra-plugins

Once the installation is done, disable strongswan from starting automatically on system boot.

systemctl disable strongswan

Login to VPN server and copy the VPN server CA certificate to the VPN client. Put the CA certificate under /etc/ipsec.d/cacerts.

Configure VPN client authentication just like you did in the server configuration.

cat /etc/ipsec.secrets

# This file holds shared secrets or RSA private keys for authentication.

# RSA private key for this host, authenticating it to any other host

# which knows the public part.

vpnsecure : EAP "P@sSw0Rd"

Define the VPN client configuration parameters.

vim /etc/ipsec.conf

conn ipsec-ikev2-vpn-client

auto=start

right=vpnsvr.example.com

rightid=vpnsvr.example.com

rightsubnet=0.0.0.0/0

rightauth=pubkey

leftsourceip=%config

leftid=vpnsecure

leftauth=eap-mschapv2

eap\_identity=%identity

Save the configuration file and restart the strongswan.

systemctl restart strongswan

Check the VPN connection status.

ipsec status

Security Associations (1 up, 0 connecting):

ipsec-ikev2-vpn-client[1]: ESTABLISHED 2 minutes ago, 10.0.2.15[vpnsecure]...192.168.43.85[vpnsvr.example.com]

ipsec-ikev2-vpn-client{1}: INSTALLED, TUNNEL, reqid 1, ESP in UDP SPIs: c80ec31a\_i cb406749\_o

ipsec-ikev2-vpn-client{1}: 10.10.7.1/32 === 0.0.0.0/0

Great. If you are using CentOS 7 as VPN client, install strongSwan as shown below and follow;

yum install strongswan strongswan-charon-nm

Copy the VPN server CA certificate to the /etc/strongswan/ipsec.d/cacerts directory.

Configure the VPN client. Replace the value of the leftid with a different username.

vim /etc/strongswan/ipsec.conf

conn ipsec-ikev2-vpn-client

auto=start

right=vpnsvr.example.com

rightid=vpnsvr.example.com

rightsubnet=0.0.0.0/0

rightauth=pubkey

leftsourceip=%config

leftid=amos

leftauth=eap-mschapv2

eap\_identity=%identity

Configure VPN Authentication details.

vim /etc/strongswan/ipsec.secrets

amos : EAP "P@sSw0Rda"

Save the configuration file and restart the strongswan.

systemctl restart strongswan

Check the VPN connection status.

strongswan status

Security Associations (1 up, 0 connecting):

ipsec-ikev2-vpn-client[1]: ESTABLISHED 1 minutes ago, 10.0.2.15[vpnsecure]...192.168.43.85[vpnsvr.example.com]

ipsec-ikev2-vpn-client{1}: INSTALLED, TUNNEL, reqid 1, ESP in UDP SPIs: c68a242b\_i cf82c13c\_o

ipsec-ikev2-vpn-client{1}: 10.10.7.2/32 === 0.0.0.0/0

Well, try to test SSH connections between the two VPN clients and you should be able to login.

Want to use Debian 10 Buster instead as strongSwan VPN Server instead? Follow the link below;

[Setup IPSEC VPN using StrongSwan on Debian 10](https://kifarunix.com/setup-ipsec-vpn-using-strongswan-on-debian-10/" \t "/home/bleach/Documents\\x/_blank)

That is all about how to configure IPSEC VPN using strongSwan on Ubuntu 18.04. Enjoy