

Assignment-4 (“VPN & IDS/IPS”)

Date: 2021-Nov-07

Submitted By: Bibek Mishra

1. Setup a VPN server in one vm. You can use openvpn for this purpose.
 - a. You should have two network interfaces one for wan and another for lan. You should set up a vpn server which listens on the WAN interface and provides a LAN interface subnets ip address to the client which connects using openvpn client.
 - b. You should create certificates files for both server and client to connect to server and export client certificates to the client vm.

VPN Server Setup on Centos 7

1- a. Answer

We have two Network Interface

WAN - 192.168.1.0/24

LAN - 10.10.1.0/8

```
[root@localhost lib]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default 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 pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:61:1c:77 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.139/24 brd 192.168.1.255 scope global noprefixroute dynamic enp0s3
        valid_lft 67932sec preferred_lft 67932sec
    inet6 fe80::f233:a532:bbba:d155/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:02:0e:ac brd ff:ff:ff:ff:ff:ff
    inet 10.10.1.1/8 brd 10.255.255.255 scope global noprefixroute enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe02:eac/64 scope link
        valid_lft forever preferred_lft forever
```

VPN setup using OpenVPN

To install OpenVPN server on Centos 7 using wget

```
sudo apt install -y epel-release
```

```
sudo apt install -y openvpn wget
```

We need Easy-RSA primarily for key management and also for web certificates.

```
wget -O /tmp/easyrsa https://github.com/OpenVPN/easy-rsa-old/archive/2.3.3.tar.gz
```

```
tar xzf /tmp/easyrsa
```

Creating a sub-directory under `/etc/openvpn` and extracting EasyRSA files over here.

```
sudo mkdir /etc/openvpn/easy-rsa
```

```
sudo cp -rf easy-rsa-old-2.3.3/easy-rsa/2.0/* /etc/openvpn/easy-rsa
```

```
[root@localhost easy-rsa]# ls
build-ca          build-key-server  list-crl          revoke-full
build-dh          build-req         openssl-0.9.6.cnf sign-req
build-inter       build-req-pass    openssl-0.9.8.cnf vars
build-key         clean-all        openssl-1.0.0.cnf whichopensslcnf
build-key-pass    inherit-inter     openssl.cnf       pkitsol
build-key-pkcs12  keys
```

Changing directory's owner to non-root sudo user

```
sudo chown bibek /etc/openvpn/easy-rsa/
```

```
[root@localhost openvpn]# ll
total 52
-rw-r--r-- 1 root root 2455 Nov  5 16:47 ca.crt
drwxr-x--- 2 root openvpn 6 Apr 21 2021 client
-rw-r--r-- 1 root root 424 Nov  5 16:47 dh2048.pem
drwxr-xr-x 3 bibek root 4096 Nov  5 16:47 easy-rsa
```

Configuring OpenVPN

We will use the server example configuration file from its documentation directory.

```
sudo cp /usr/share/doc/openvpn-2.4.11/sample/sample-config-files/server.conf /etc/openvpn/
```

```
sudo nano /etc/openvpn/server.conf and make following changes
```

- To listen at WAN address

```
local 192.168.1.139 -- ip address of centos 7 router (WAN)
```

- Default port

```
port 1194
```

- I have enabled both tcp and udp protocol

proto tcp

proto udp

- To create routed IP tunnel

dev tun

- Default Client and server certificate & key names

ca ca.crt

cert server.crt

key server.key

- Default Diffie-Hellman parameter name

dh dh2048.pem

- Network topology

topology subnet

- To give client address

server 10.10.1.0 255.255.255.0

- Push routes to the client to allow it to reach each other private subnets behind the server

push "route 192.168.10.0 255.255.255.0"

- DNS servers

push "dhcp-option DNS 8.8.8.8"

push "dhcp-option DNS 8.8.4.4"

- To allow different clients to see each other

client-to-client

- For extra security beyond that provided by SSL/TLS, create an "HMAC firewall" (block DoS attack and UDP port flooding)

tls-crypt myvpn.tlsauth

- For non-windows system

user nobody

group nobody

- To append log at specific location

log /var/log/openvpn.log

- To notify client when the server restarts

explicit-exit-notify 1

- For tls web client authentication

remote-cert-eku "TLS Web Client Authentication"

- For user password authentication

plugin /usr/lib64/openvpn/plugins/openvpn-plugin-auth-pam.so openvpn

- Generating static encryption key

sudo openvpn --genkey --secret /etc/openvpn/myvpn.tlsauth

1- b. Answer

Creating certificates files for both server and client to connect to server

- Creating keys directory where Easy-RSA will store any keys and certs we generate

sudo mkdir /etc/openvpn/easy-rsa/keys

- Default certificate variables are set in vars file in /etc/openvpn/easy-rsa

sudo nano /etc/openvpn/easy-rsa/vars

Leaving others as default change the following parameters as per required

export KEY_COUNTRY="NP"

export KEY_PROVINCE="KTM"

export KEY_CITY="Kathmandu"

export KEY_ORG="LFTechnology"

export KEY_EMAIL="root@example.com"

export KEY_EMAIL=root@example.com

export KEY_CN=192.168.1.139

export KEY_NAME="EasyRSA"

export KEY_OU=LFTechnology

Save and exit

- To start generating keys, move to easy-rsa directory and source in the new variables

cd /etc/openvpn/easy-rsa

source ./vars

- Clean any keys and certificates already in the folder

./clean-all

- Build certificate authority. We have already set variables in the vars file, so we can press ENTER to accept the defaults for each one

./build-ca - *this script generates ca.key used to sign your server and client's certificates*

- Creating key and certificate for the server

./build-key-server server

- Creating diffie helmen key exchange file

./build-dh - this can take few minutes to complete

- Now copy the server keys and certificates from **keys** directory to **openvpn** directory

cd /etc/openvpn/easy-rsa/keys

sudo cp dh2048.pem ca.crt server.crt server.key /etc/openvpn

```
[root@localhost openvpn]# ls
ca.crt  dh2048.pem ipp.txt      openvpn-status.log  server.conf  server.key
client  easy-rsa  myvpn.tlsauth  server              server.crt
[root@localhost openvpn]#
```

Generating client keys

- We called it client, but you can give more descriptive name

cd /etc/openvpn/easy-rsa

./build-key client

```
[root@localhost easy-rsa]# cd keys/
[root@localhost keys]# ls
01.pem          ca.key          dh2048.pem      index.txt.old   server.csr
02.pem          client.crt      index.txt       serial          server.key
bibek@192.168.1.142 client.csr      index.txt.attr  serial.old
ca.crt          client.key      index.txt.attr.old  server.crt
[root@localhost keys]#
```

- Copy versioned OpenSSL configuration file to versionless name to load configuration

cp /etc/openvpn/easy-rsa/openssl-1.0.0.cnf /etc/openvpn/easy-rsa/openssl.cnf

```
[root@localhost easy-rsa]# ls
build-ca          build-key-server  list-crl         revoke-full
build-dh          build-req         openssl-0.9.6.cnf  sign-req
build-inter       build-req-pass    openssl-0.9.8.cnf  vars
build-key         clean-all        openssl-1.0.0.cnf  whichopensslcnf
build-key-pass    inherit-inter     openssl.cnf
build-key-pkcs12  keys              pkitool
[root@localhost easy-rsa]#
```

Giving instructions to OpenVPN about where to send incoming web traffic (ROUTING)

- Adding openvpn service permanently to external active zone

sudo firewall-cmd --zone=external --add-service openvpn --permanent

- Adding masquerade to all future instances with --permanent

sudo firewall-cmd --permanent --add-masquerade

- Check that the masquerade was added correctly

sudo firewall-cmd --query-masquerade - output must be yes

```
[root@localhost keys]# firewall-cmd --query-masquerade
yes
[root@localhost keys]#
```

- Forwarding routing to OpenVPN subnet
- Creating variable SHARK which will represent the primary network interface

```
SHARK=$(ip route get 8.8.8.8 | awk 'NR==1 {print $(NF-2)}')
```

- Using SHARK variable to permanently add the routing rule to our subnet

```
sudo firewall-cmd --permanent --direct --passthrough ipv4 -t nat -A POSTROUTING -s
10.10.1.0/24 -o $$SHARK -j MASQUERADE
```

- Reloading firewall-cmd

```
sudo firewall-cmd --reload
```

- We have to enable `ip_forwarding=1`

We have done it previously permanently, configuring Centos as a router

- Restart Network service

```
sudo systemctl restart network
```

Now we are ready to start **openvpn service**

```
sudo systemctl -f enable openvpn@server.service
```

```
sudo systemctl start openvpn@server.service
```

```
sudo systemctl status openvpn@server.service
```

```
[root@localhost keys]# systemctl status openvpn@server.service
● openvpn@server.service - OpenVPN Robust And Highly Flexible Tunneling Applicat
ion On server
   Loaded: loaded (/usr/lib/systemd/system/openvpn@.service; enabled; vendor pre
set: disabled)
   Active: active (running) since Fri 2021-11-05 22:58:22 +0545; 58min ago
     Main PID: 6482 (openvpn)
    Status: "Initialization Sequence Completed"
      CGroup: /system.slice/system-openvpn.slice/openvpn@server.service
              └─6482 /usr/sbin/openvpn --cd /etc/openvpn/ --config server.conf
                └─6485 /usr/sbin/openvpn --cd /etc/openvpn/ --config server.conf

Nov 05 22:58:22 localhost.localdomain systemd[1]: Starting OpenVPN Robust And...
Nov 05 22:58:22 localhost.localdomain systemd[1]: Started OpenVPN Robust And ...
Hint: Some lines were ellipsized, use -l to show in full.
```

To transfer client certificate to client machine, I used rsync command

- The keys to transfer to client machine are **ca.crt**, **client.crt**, **client.key**(all three are in keys directory) & **myvpn.tlsauth** (is in openvpn directory)
- Change directory path to keys and use rsync command

cd /etc/openvpn/easy-rsa/keys

sudo rsync ca.crt client.crt client.key ../../myvpn bibek@192.168.1.142:/home/bibek/openclient

And provided password for bibek user of 192.168.1.142 server

```
bibek@bibek-lf:~/openclient$ pwd
/home/bibek/openclient
bibek@bibek-lf:~/openclient$ ls
ca.crt client.crt client.key client.ovpn myvpn.tlsauth
bibek@bibek-lf:~/openclient$
```

2. Create another vm with installing openvpn client and you should create an openvpn client connect file(with extension .ovpn) providing the certificates. You should be able to connect to the vpn server and get an ip address from the LAN subnet that you assigned in the first vm.

My other VM is ubuntu.

I created client.ovpn in the same directory where I have transferred the client keys so I have just mentioned the certificate names(no need to give path)

sudo nano client.ovpn

Add add following lines

```
client
tls-client
ca ca.crt
cert client.crt
key client.key
tls-crypt myvpn.tlsauth
remote-cert-eku "TLS Web Server Authentication"
proto tcp
remote 192.168.1.139 1194 udp
dev tun
topology subnet
pull
user nobody
group nobody
auth-user-pass
```

```

client
tls-client
ca ca.crt
cert client.crt
key client.key
tls-crypt myvpn.tlsauth
remote-cert-eku "TLS Web Server Authentication"
proto tcp
remote 192.168.1.139 1194 udp
dev tun
topology subnet
pull
user nobody
group nobody
auth-user-pass

```

- To enable user-password authentication
- We need to create openvpn file in **/etc/pam.d/** directory in centos server where openVPN server is

sudo vi /etc/pam.d/openvpn

And add the following lines

```

auth    required    pam_unix.so  shadow nodelay
account required    pam_unix.so

```

```

auth    required    pam_unix.so  shadow nodelay
account required    pam_unix.so
~

```

Now install openvpn-client to ubuntu server

sudo apt install -y openvpn

- To connect to openvpn server

sudo openvpn --config client.ovpn

- Ip address of ubuntu server before connecting to openvpn


```

bibek@bibek-lf:~/openclient$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default 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 group default qlen 1000
    link/ether 08:00:27:75:a8:15 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.142/24 brd 192.168.1.255 scope global dynamic noprefixroute enp0s3
        valid_lft 55396sec preferred_lft 55396sec
    inet6 fe80::e9f4:8400:a73a:c0c/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
bibek@bibek-lf:~/openclient$

```

- Give username and password

```

Sun Nov  7 01:44:07 2021 WARNING: file 'myvpn.tlsauth' is not readable
Sun Nov  7 01:44:07 2021 OpenVPN 2.4.7 x86_64-pc-linux-gnu [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] built on Jun 11 2021
Sun Nov  7 01:44:07 2021 library versions: OpenSSL 1.1.1g
Enter Auth Username: bibek
Enter Auth Password: *****
Sun Nov  7 01:44:17 2021 WARNING: you are using user/group persist-tun -- this may cause restarts to fail
Sun Nov  7 01:44:17 2021 WARNING: you are using user/group persist-key -- this may cause restarts to fail
Sun Nov  7 01:44:17 2021 TCP/UDP: Preserving recently

```

- Successfully connected

```

Sun Nov  7 01:44:17 2021 [server] Peer Connection Initiated with [AF_INET]192.168.1.139:1194
Sun Nov  7 01:44:18 2021 TUN/TAP device tun0 opened
Sun Nov  7 01:44:18 2021 /sbin/ip link set dev tun0 up mtu 1500
Sun Nov  7 01:44:18 2021 /sbin/ip addr add dev tun0 10.10.1.2/24 broadcast 10.10.1.255
Sun Nov  7 01:44:18 2021 GID set to nobody
Sun Nov  7 01:44:18 2021 UID set to nobody
Sun Nov  7 01:44:18 2021 Initialization Sequence Completed

```

- New tunneling IP address after VPN connection

```

bibek@bibek-lf:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defau
lt 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 g
roup default qlen 1000
    link/ether 08:00:27:75:a8:15 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.142/24 brd 192.168.1.255 scope global dynamic noprefixroute
enp0s3
        valid_lft 55135sec preferred_lft 55135sec
    inet6 fe80::e9f4:8400:a73a:c0c/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
11: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel sta
te UNKNOWN group default qlen 100
    link/none
    inet 10.10.1.2/24 brd 10.10.1.255 scope global tun0
        valid_lft forever preferred_lft forever
    inet6 fe80::d7f9:4de:a0be:deea/64 scope link stable-privacy
        valid_lft forever preferred_lft forever
bibek@bibek-lf:~$

```

- Ping result to host only network

```

11: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel sta
te UNKNOWN group default qlen 100
    link/none
    inet 10.10.1.2/24 brd 10.10.1.255 scope global tun0
        valid_lft forever preferred_lft forever
    inet6 fe80::d7f9:4de:a0be:deea/64 scope link stable-privacy
        valid_lft forever preferred_lft forever
bibek@bibek-lf:~$ ping 10.10.1.1
PING 10.10.1.1 (10.10.1.1) 56(84) bytes of data.
64 bytes from 10.10.1.1: icmp_seq=1 ttl=64 time=0.963 ms
64 bytes from 10.10.1.1: icmp_seq=2 ttl=64 time=1.46 ms
64 bytes from 10.10.1.1: icmp_seq=3 ttl=64 time=0.930 ms
^C
--- 10.10.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.930/1.118/1.461/0.242 ms
bibek@bibek-lf:~$

```

In this way OpenVPN is configured successfully and the client machine gets connected.

Q-3. For IDS/IPS I have used Suricata Application

To install **suricata**
yum install epel-release yum-plugin-copr
yum copr enable @oisf/suricata-6.0
yum install suricata

Check network interface name to provide to suricata for inspecting
ip a

To configure Suricata

We have configuration file located at **/etc/suricata/suricata.yaml**

vi /etc/suricata/suricata.yaml

Edit HOME_NET

HOME_NET variable should include, in most scenarios, the IP address of the monitored interface and all the local networks in use

Capture setting:

af-packet:

- interface: enp0s3

cluster-id: 99

cluster-type: cluster_flow

defrag: yes

use-mmap: yes

tpacket-v3: yes

We need to disable packet offload features on the network interface on which Suricata is listening.

ethtool -K enp0s3 gro off lro off

Verify this

ethtool -k enp0s3 | grep large

```
[root@localhost bibek]# ethtool -k enp0s3 | grep large
large-receive-offload: off [fixed]
```

We can **edit suricata rules** located at **/usr/share/suricata/rules/**

Adding new sample rules

vim /usr/share/suricata/rules/test.rules

Add following line

*alert http any any -> any any (msg:"Do not read gossip during work";
content:"Scarlett"; nocase; classtype:policy-violation; sid:1; rev:1;)*

Save and exit

```
[root@localhost rules]# ls
2100498          files.rules          ntp-events.rules
app-layer-events.rules  http-events.rules   smb-events.rules
decoder-events.rules  ipsec-events.rules  smtp-events.rules
dhcp-events.rules     kerberos-events.rules  stream-events.rules
dnp3-events.rules     modbus-events.rules  test.rules
dns-events.rules      nfs-events.rules     tls-events.rules
[root@localhost rules]#
```

Add rules name in suricata.yaml

vim /usr/share/suricata/rules/suricata.yaml

Find rule-files

And under it write

- /usr/share/suricata/rules/test.rules

fire Suricata in PCAP live mode by executing

suricata -D -c /etc/suricata/suricata.yaml -i enp0s3

```
[root@localhost rules]# suricata -D -c /etc/suricata/suricata.yaml -i enp0s3
7/11/2021 -- 18:14:49 - <Notice> - This is Suricata version 6.0.3 RELEASE running in SYSTEM mode
```

fetches the ET Open ruleset

suricata-update

```
[root@localhost bibek]# suricata-update
7/11/2021 -- 18:41:12 - <Info> -- Using data-directory /var/lib/suricata.
7/11/2021 -- 18:41:12 - <Info> -- Using Suricata configuration /etc/suricata/suricata.yaml
7/11/2021 -- 18:41:12 - <Info> -- Using /usr/share/suricata/rules for Suricata provided rules.
7/11/2021 -- 18:41:12 - <Info> -- Found Suricata version 6.0.3 at /sbin/suricata.
7/11/2021 -- 18:41:12 - <Info> -- Loading /etc/suricata/suricata.yaml
7/11/2021 -- 18:41:12 - <Info> -- Disabling rules for protocol http2
7/11/2021 -- 18:41:12 - <Info> -- Disabling rules for protocol modbus
7/11/2021 -- 18:41:12 - <Info> -- Disabling rules for protocol enip
7/11/2021 -- 18:41:12 - <Info> -- Disabling rules for protocol dnp3
7/11/2021 -- 18:41:12 - <Info> -- No sources configured, will use Emerging Threats Open
7/11/2021 -- 18:41:12 - <Info> -- Checking https://rules.emergingthreats.net/open/suricata-6.0.3/emerging.rules.tar.gz.md5.
```

```

7/11/2021 -- 18:41:18 - <Info> -- Loading distribution rule file /usr/share/suricata/rules/tls-events.rules
7/11/2021 -- 18:41:19 - <Info> -- Ignoring file rules/emerging-deleted.rules
7/11/2021 -- 18:41:29 - <Info> -- Loaded 31207 rules.
7/11/2021 -- 18:41:30 - <Info> -- Disabled 14 rules.
7/11/2021 -- 18:41:30 - <Info> -- Enabled 0 rules.
7/11/2021 -- 18:41:30 - <Info> -- Modified 0 rules.
7/11/2021 -- 18:41:30 - <Info> -- Dropped 0 rules.
7/11/2021 -- 18:41:31 - <Info> -- Enabled 131 rules for flowbit dependencies.
7/11/2021 -- 18:41:31 - <Info> -- Backing up current rules.
7/11/2021 -- 18:42:48 - <Info> -- Writing rules to /var/lib/suricata/rules/suricata.rules: total: 31207; enabled: 23841; added: 0; removed 0; modified: 0
7/11/2021 -- 18:42:51 - <Info> -- Writing /var/lib/suricata/rules/classification.config
7/11/2021 -- 18:42:51 - <Info> -- No changes detected, exiting.

```

Tail the Suricata alert logs on Suricata host to see what is happening;

tail -f /var/log/suricata/fast.log

```

[root@localhost rules]# tail -f /var/log/suricata/fast.log
11/07/2021-18:20:25.424799  [**] [1:0:0] Do not read gossip during work [**] [Classification: (null)] [Priority: 3] {TCP} 35.232.111.17:80 -> 192.168.1.142:48632
11/07/2021-18:20:25.710640  [**] [1:0:0] Do not read gossip during work [**] [Classification: (null)] [Priority: 3] {TCP} 35.232.111.17:80 -> 192.168.1.142:48632
11/07/2021-18:20:25.710648  [**] [1:0:0] Do not read gossip during work [**] [Classification: (null)] [Priority: 3] {TCP} 35.232.111.17:80 -> 192.168.1.142:48632
11/07/2021-18:20:25.711017  [**] [1:0:0] Do not read gossip during work [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.1.142:48632 -> 35.232.111.17:80
11/07/2021-18:20:25.711473  [**] [1:0:0] Do not read gossip during work [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.1.142:48632 -> 35.232.111.17:80
11/07/2021-18:20:25.722485  [**] [1:0:0] Do not read gossip during work [**] [Classification: (null)] [Priority: 3] {TCP} 35.232.111.17:80 -> 192.168.1.142:48632

```