OPENVPN

I)Các kiến thức cần nắm

1) Giới thiệu về giao thức SSL

- Giao thức SSL trong mô hình OSI được đặt giữa tầng vận chuyển (Transport layer) và tầng ứng dụng (Application Layer). SSL được thiết kế như một giao thức riêng cho vấn đề bảo mật có thể hỗ trợ cho rất nhiều ứng dụng . Giao thức SSL hoạt động bên trên TCP/IP và bên dưới các giao thức tầng ứng dụng như **HTTP, IMAP, FTP.** SSL không phải là một giao thức đơn lẻ, mà là một tập các thủ tục đã được chuẩn hóa để thực hiện các nhiệm vụ sau:

+) Xác thực Server : Cho phép người dùng sử dụng xác thực được server muốn kết nối tới. Lúc này phía Client sẽ sử dụng các kỹ thuật mã hóa công khai để chắc chắn rằng Certificate và Public ID của server là các giá trị được cấp phát bởi CA (certificate authority) trong danh sách các CA đáng tin cậy của client .Điều này rất quan trọng với người dùng . Ví dụ như khi gửi mã số Credit Card qua mạng thì người dùng thực sự muốn kiểm tra liệu server sẽ nhận thông tin này có đúng là server mà hộ định gửi đến không.

+) Xác thực Client: Cho phép phía server xác thực được người dùng muốn kết nối. Phía server cũng sử dụng các kỹ thuật mã hóa công khai để kiểm tra xem certificate và public ID của client có giá trị hay không và được cấp phát bởi một CA trong danh sách các CA đáng tin cậy của Server không. Điều này rất quan trọng đối với các nhà cung cấp. Ví dụ như khi một ngân hàng muốn gửi các thông tin tài chính mang tính bảo mật tới khách hàng thì họ rất muốn kiểm tra định danh của người của người nhận.

+) Mã hóa kết nối: Tất cả các thông tin trao đổi giữa client và server được mã hóa trên đường truyền nhằm nâng cao khả năng bảo mật.

Cấu trúc của SSL:

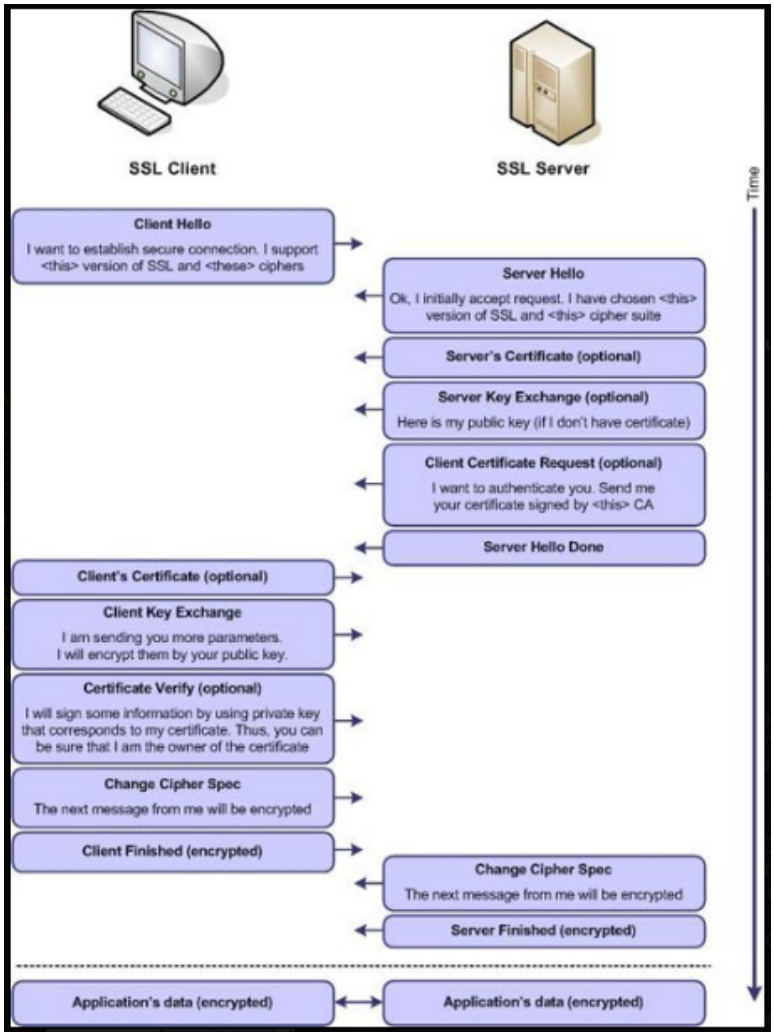
- **Handshake protocol:** được sử dụng để khởi tạo phiên SSL giữa Client và Server, nhờ giao thức này các bên sẽ xác thực lẫn nhau và thỏa thuận các tham số cho phiên làm việc sẽ được thiết lập.

- **SSL Alert protocol :** sử dụng để mang các thông điệp của phiên liên quan tới việc trao đổi dữ liệu và hoạt động của các giao thức.

- **Change Cipher Spec:** chứa 1 thông điệp mang giá trị 1 giúp chuyển trạng thái của 1 phiên từ " đang chờ" sang "bền vững".

- **SSL Record Layer**: sử dụng để trao đổi tất cả các kiểu dữ liệu trong 1 phiên, bao gồm các thông điệp, dữ liệu của các giao thức SSL khác và dữ liệu của ứng dụng. SSL Record protocol liên quan đến việc bảo mật và đảm bảo toàn vẹn dữ liệu , mục đích là thu nhận những thông điệp mà ứng dụng chuẩn bị gửi, phân mảnh dữ liệu cần truyền, đóng gói bổ sung header tạo thành 1 đống bản ghi được mã hóa và có thể truyền bằng giao thức TCP.

Hoạt động của SSL:



Khi một kết nối được thiết lập sử dụng SSL/TLS ví dụ sử dụng giao thức HTTPS(cổng mặc định 443) một thông điệp(messages) sẽ được trao đổi giữa các

client .quá trình bắt đầu được kết nối với server.Đầu tiên trao đổi các messages được gọi quá trình bắt tay “handshake “. Quá trình được trao đổi như sau:

Bước 1: Client sẽ gửi cho server số phiên bản SSL đang dùng, các tham số của thuật toán mã hoá, dữ liệu được tạo ra ngẫu nhiên (đó chính là digital signature) và một số thông tin khác mà server cần để thiết lập kết nối với client.

Bước 2: Server gửi cho client số phiên bản SSL đang dùng, các tham số của thuật toán mã hoá, dữ liệu được tạo ra ngẫu nhiên và một số thông tin khác mà client cần để thiết lập kết nối với server. Ngoài ra server cũng gửi certificate của nó đến client, và yêu cầu certificate của client nếu cần.

Bước 3: Client sử dụng một số thông tin mà server gửi đến để xác thực server. Nếu như server không được xác thực thì người sử dụng sẽ được cảnh báo và kết nối không được thiết lập. Còn nếu như xác thực được server thì phía client sẽ thực hiện tiếp
Bước 4: Sử dụng tất cả các thông tin được tạo ra trong giai đoạn bắt tay ở trên, client (cùng với sự cộng tác của server và phụ thuộc vào thuật toán được sử dụng) sẽ tạo ra premaster secret cho phiên làm việc, mã hoá bằng khoá công khai (public key) mà server gửi đến trong certificate ở bước 2, và gửi đến server.

Bước 5: Nếu server có yêu cầu xác thực client, thì phía client sẽ đánh dấu vào phần thông tin riêng chỉ liên quan đến quá trình “bắt tay” này mà hai bên đều biết. Trong trường hợp này, client sẽ gửi cả thông tin được đánh dấu và certificate của mình cùng với premaster secret đã được mã hoá tới server.

Bước 6: Server sẽ xác thực client. Trường hợp client không được xác thực, phiên làm việc sẽ bị ngắt. Còn nếu client được xác thực thành công, server sẽ sử dụng khoá bí mật (private key) để giải mã premaster secret, sau đó thực hiện một số bước để tạo ra master secret.

Bước 7: Client và server sẽ sử dụng master secret để tạo ra các session key, đó chính là các khoá đối xứng được sử dụng để mã hoá và giải mã các thông tin trong phiên làm việc và kiểm tra tính toàn vẹn dữ liệu.

Bước 8: Client sẽ gửi một lời nhắn đến server thông báo rằng các message tiếp theo sẽ được mã hoá bằng session key. Sau đó nó gửi một lời nhắn đã được mã hoá để thông báo rằng phía client đã kết thúc giai đoạn "bắt tay".

Bước 9: Server cũng gửi một lời nhắn đến client thông báo rằng các message tiếp theo sẽ được mã hoá bằng session key. Sau đó nó gửi một lời nhắn đã được mã hoá để thông báo rằng server đã kết thúc giai đoạn "bắt tay".

Bước 10: Lúc này giai đoạn "bắt tay" đã hoàn thành, và phiên làm việc SSL bắt đầu. Cả hai phía client và server sẽ sử dụng các session key để mã hoá và giải mã thông tin trao đổi giữa hai bên, và kiểm tra tính toàn vẹn dữ liệu

2) Giới thiệu về OpenVPN

OpenVPN hoạt động theo mô hình Client-server, tạo ra các giao diện ethernet ảo , cho phép mã hóa và xác thực mọi dữ liệu (Mail, Web, FTP, Text..) khi truyền qua giao diện ảo này. OpenVPN là phần mềm mã nguồn mở hoạt động trên môi trường như Windows, Linux, Unix…

Cấu trúc của bộ chương trình OpenVPN:

- ***Thành phần điều khiển kết nối và trao đổi khóa***

+) Tạo ra một VPN Daemon để khởi tạo và vận hành OpenVPN

+) Tạo ra một đường hầm dựa trên TCP

+) Điều khiển các phiên kết nối: nếu là khóa tĩnh thì đọc khóa từ file, nếu là SSL/TLS thì tạo ra một kênh kết nối SSL dựa trên kênh TCP đã tạo ra để thực hiện việc trao đổi khóa trên kênh SSL đó.

- ***Thành phần xử lý dữ liệu*** **:** thực hiện các công việc như nén/giải nén, mã hóa/giải mã, xác thực/kiểm tra, gửi/nhận gói dữ liệu trên kênh TCP

- ***Thành phần tương tác với nhân của hệ điều hành để gửi nhận dữ liệu***

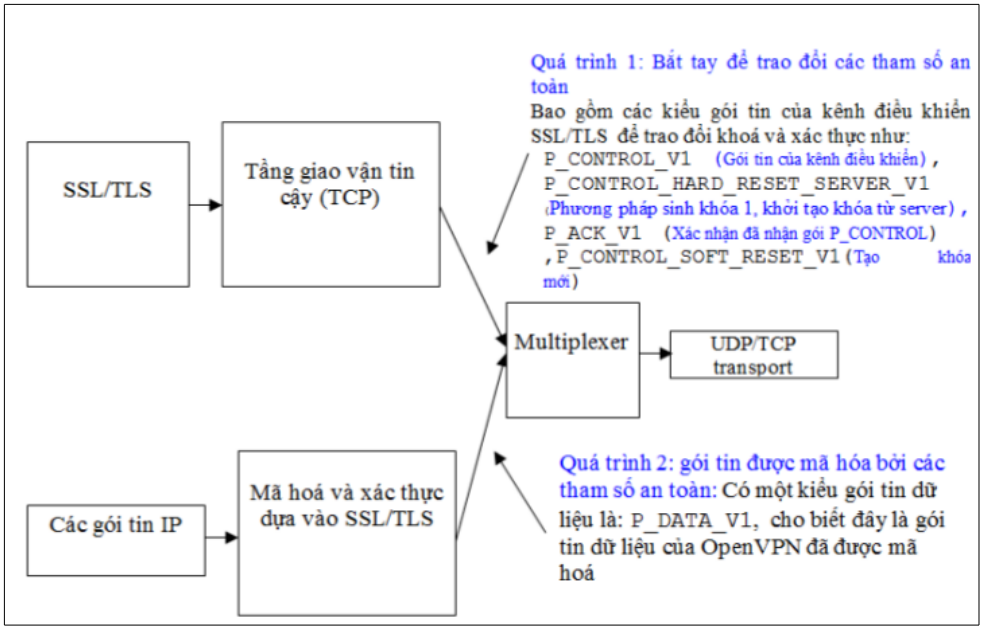
+) Đây là trình điều khiển driver của OpenVPN. Trên Linux, driver này đã được cung sẵn trong nhân. Trên Windows chúng ta phải biên dịch và cài đặt driver này trong thư mục tap-driver của thư mục OpenVPN

+) Driver này có nhiệm vụ nhận gói dữ liệu từ card mạng vật lý đưa lên không gian người dùng để OpenVPN daemon xử lý (nén/giải nén , mã hóa/giải mã, xác thức/kiểm tra, phân luồng).

+) Sau khi xử lý xong thì gửi trả về cho Driver này thực hiện tạo gói tin Ethernet để truyển đi.

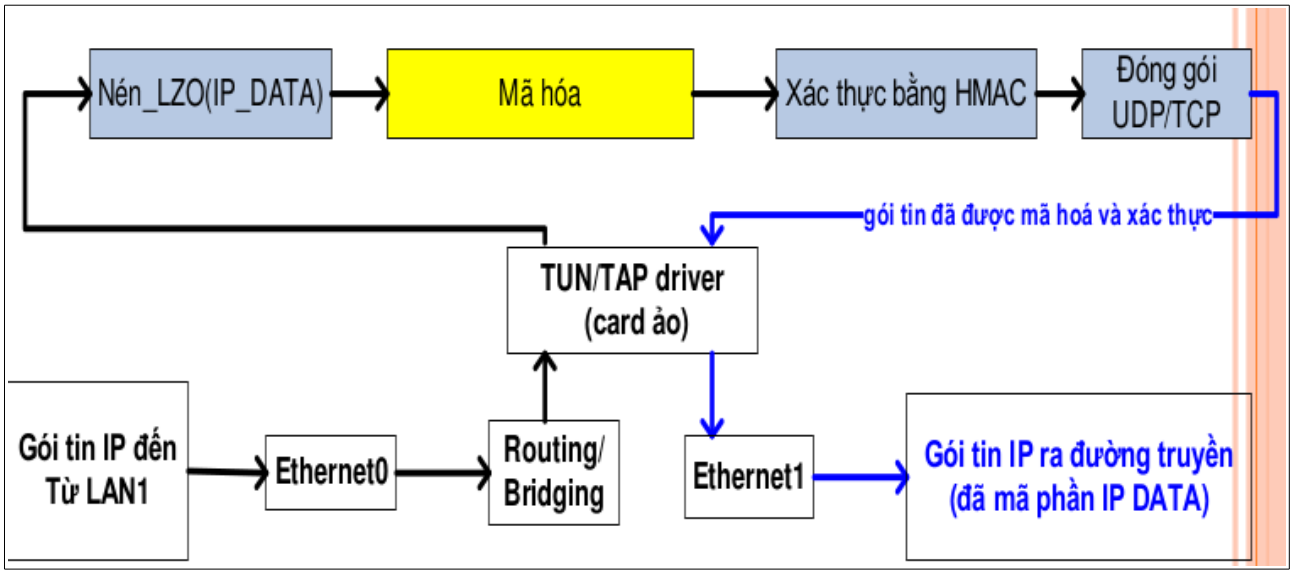
Cơ chế hoạt động của OpenVPN:

OpenVPN kết hợp giữ một phiên liên lạc SSL/TLS dùng cho việc xác thực, trao đổi khóa với việc mã hóa dòng dữ liệu, nó cung cấp kết nối SSL/TLS trong tầng giao vận tin cậy. Tiếp sau đó là quá trình mã hóa gói tin và truyền tải dựa trên giao thức UDP hoặc TCP. Các gói tin IP ban đầu, sau khi đã được mã hóa và ký với một HMAC, sẽ được gửi qua đường hầm (Tunel) dựa trên giao thức UDP hoặc TCP.

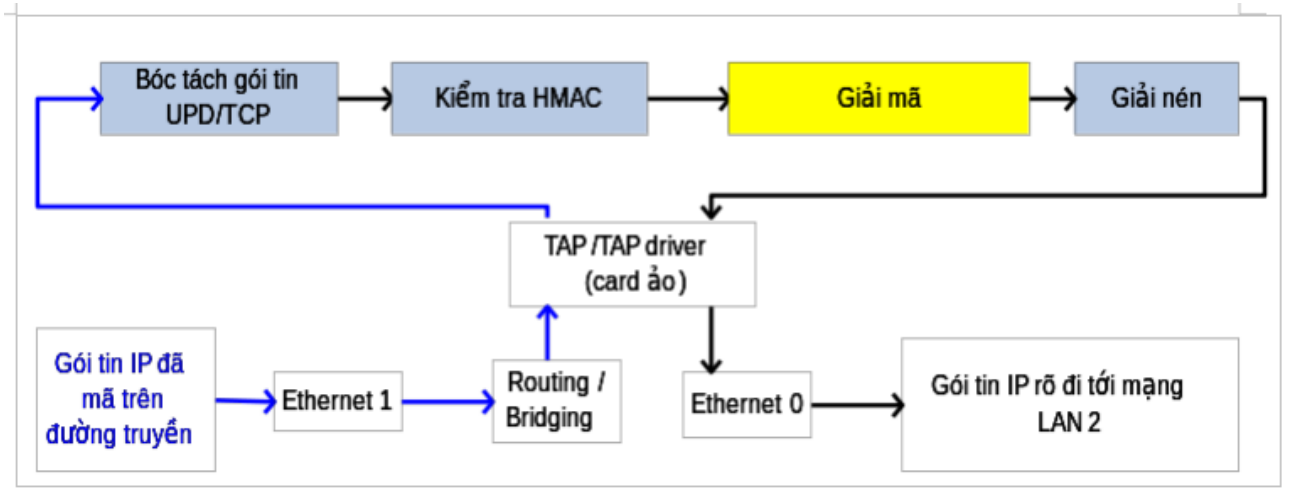


Quá trình mã hóa và giải mã trong OpenVPN

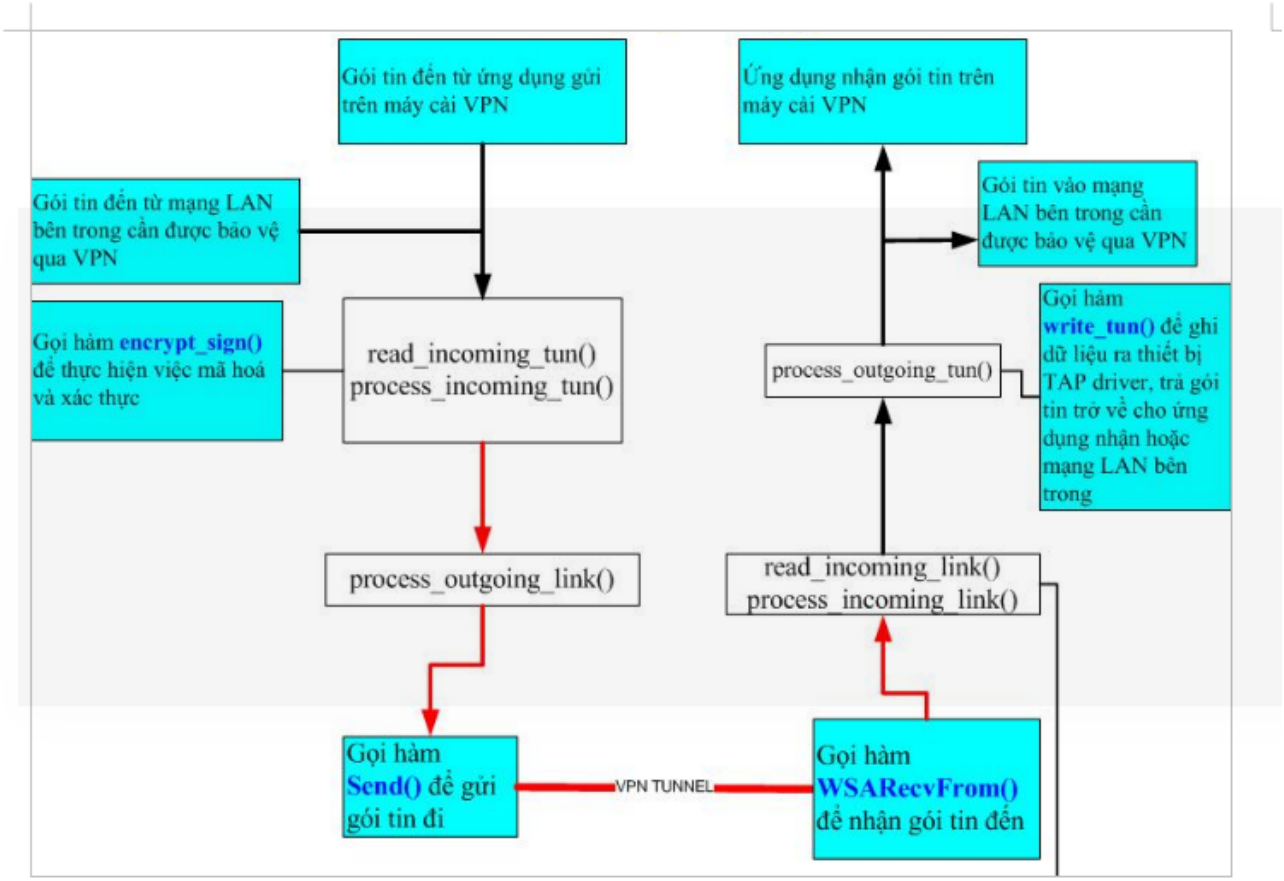
- Quá trình truyền nhận và bắt gói tin được thực hiện trong card mạng ảo TUN/TAP trong KernelSpace.
- Quá trình nén dữ liệu, mã hóa, xác thực và đóng gói tin theo UDP/TCP được thực hiện trong UserSpace.



- Quá trình nhận gói tin, giải nén, giải mã, loại bỏ phần đóng gói gói tin theo UDP/TCP được thực hiện ngược lại như sau.



Quy trình xử lý gói tin trên OpenVPN



Bên gửi: Gói tin gửi từ mạng LAN (hoặc từ máy hiện tại) sau khi qua xử lý routing sẽ được gửi đến TAP driver để đưa lên tầng ứng dụng cho VPN daemon xử lý. Ở đây, VPN daemon đầu tiên sẽ gọi hàm process_incoming_tun để nhận và xử lý. Hàm này sẽ gọi hàm process_ipv4_header để cấp phát VPN header (Ethernet header), sau đó gọi đến hàm encrypt_sign để thực hiện nén, mã và xác thực bằng HMAC. Trong hàm encrypt_sign gọi đến hàm link_socket_get_outgoing_addr để thiết lập địa chỉ nơi gửi đến (UDP socket). Tiếp đó hàm process_outgoing_link sẽ gọi đến hàm send() của hệ thống để gửi gói tin ra card mạng thật. Lúc này gói tin đã được mã hoá.

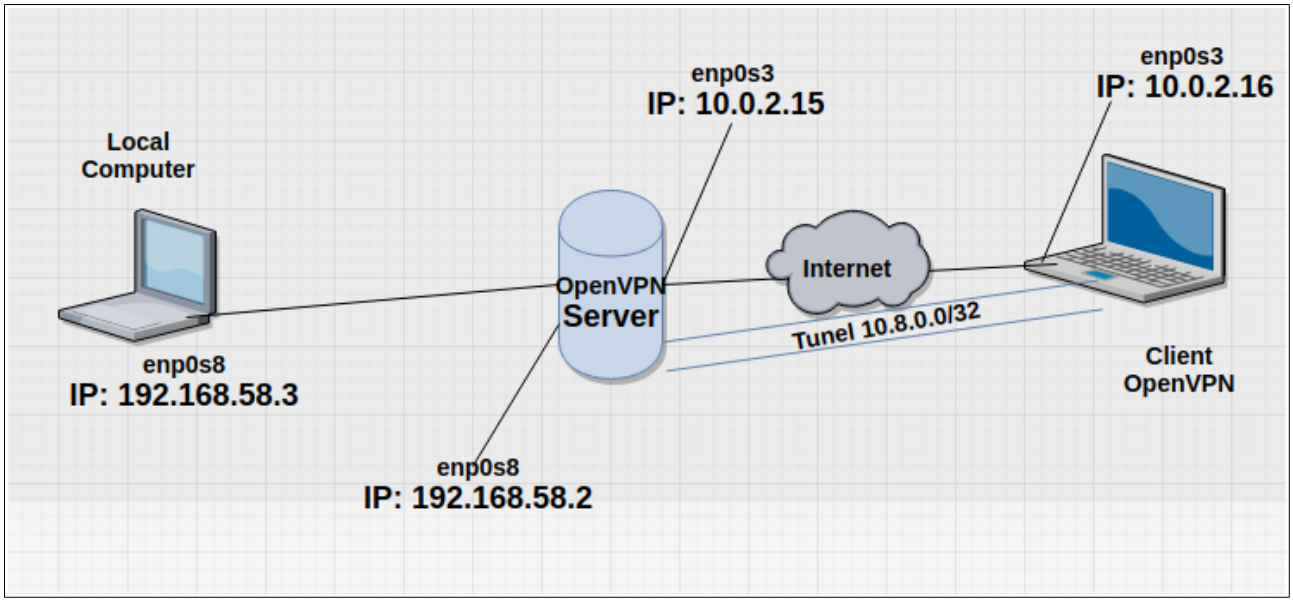
Bên nhận: Hàm hệ thống WSARecvFrom() thực hiện việc nhận dữ liệu. Hàm read_incoming_link () xử lý dữ liệu đến sau đó chuyển cho hàm process_incoming_link() thực hiện việc giải mã, xác thực và giải nén. Sau đó sẽ gọi hàm process_outing_tun() để chuyển về cho ứng dụng hoặc sang mạng LAN bên kia. Lúc này gói tin đã được giải mã.

Các hàm process_incoming_tun và hàmprocess_outgoing_tun đều gọi đến process_ipv4_header. Trong OpenVPN hàm process_ipv4_header chỉ đóng gói gói tin theo Ethernet header (TAP device). Hàm process_ipv4_header gọi đến hàm is_ipv4(), hàm này thực hiện việc ép kiểu dữ liệu thô trong buffer theo cấu trúc openvpn_ethhdr và đưa con trỏ dữ liệu qua phần header để mã hoá phần data mà không mã phần header. Hàm process_io trong openvpn.c thực hiện vòng lặp để đọc/ghi và xử lý dữ liệu vào ra.

II) Triển khai demo

1) Thiết lập Client-to-Site

- Yêu cầu các user login bằng Username + Password
- Mô hình triển khai:



***)Cấu hình**

Trên Server.

- Trước tiên tiến hành cài đặt OpenVPN:

#sudo apt-get update

#sudo apt-get install openvpn easy-rsa

Cài đặt CA Directory:

#make-cadir /etc/openvpn/openvpn-ca

- Di chuyển tới thư mục vừa khởi tạo sử dụng các câu lệnh sau

#vim vars

Cấu hình thông tin để tạo Certificate

export KEY_COUNTRY="VN"

export KEY_PROVINCE="HN"

export KEY_CITY="HN"

export KEY_ORG="VCCLOUD"

export KEY_EMAIL="me@myhost.mydomain"

export KEY_OU="VCCORP"

Chỉnh sửa Key_name

export KEY_NAME="server"

- Build CA

#cd /etc/openvpn/openvpn-ca

#source vars

Sẽ có một thông báo như sau

Output

NOTE: If you run ./clean-all, I will be doing a rm -rf on /home/hannv/openvpn-ca/keys

#./clean-all

Bây giờ ta sẽ tạo root CA bằng câu lệnh sau:

#./build-ca

Generating a 2048 bit RSA private key

.....+++

.....+++

writing new private key to 'ca.key'

You are about to be asked to enter information that will be incorporated

into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

Country Name (2 letter code) [VN]:

State or Province Name (full name) [VN]:

Locality Name (eg, city) [HN City]:

Organization Name (eg, company) [VCCLOUD]:

Organizational Unit Name (eg, section) [VCCORP]:

Common Name (eg, your name or your server's hostname) [VCCLOUD CA]:

Name [server]:

Email Address [admin@email.com]:

Sau đó chúng ta sẽ tiến hành tạo server keys,và đặt tên là “server” các bước như sau:

#./build-key-server server

Generating a 1024 bit RSA private key

```
.....++++++
.............................++++++
writing new private key to 'server.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [VN]:
State or Province Name (full name) [HN]:
Locality Name (eg, city) [HN]:
Organization Name (eg, company) [VCCORP]:
Organizational Unit Name (eg, section) [VCCLOUD]:
Common Name (eg, your name or your server's hostname) [server]:
Name [changeme]:
Email Address [mail@host.domain]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from /etc/openvpn/easy-rsa/openssl.cnf
Check that the request matches the signature
```

```
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'VN'
stateOrProvinceName   :PRINTABLE:'HN'
localityName          :PRINTABLE:'HN'
organizationName      :PRINTABLE:'VCCORP'
organizationalUnitName:PRINTABLE:'VCCLOUD'
commonName            :PRINTABLE:'server'
name                  :PRINTABLE:'changeme'
emailAddress          :IA5STRING:'mail@host.domain'

Certificate is to be certified until Feb  3 09:26:00 2024 GMT (3650 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

Tiếp theo ta cần tạo ra một keys Diffe-Hellman đủ mạnh để sử dụng trong quá trình trao đổi khóa.

#./build-dh

```
Generating DH parameters, 1024 bit long safe prime, generator 2
This is going to take a long time
.....+.....+.....
+.....+.....
```

Bây giờ ta đã có đủ các Certificates và Keys mà Server cần ta cần coppy chúng tới thư mục /etc/openvpn/ bằng câu lệnh sau:

 # cd /etc/openvpn/easy-rsa/keys/

 # cp server.key server.crt ca.crt dh1024.pem /etc/openvpn

Sau khi chuyển các certificates, keys đến thư mục /etc/openvpn ta tiến hành cấu hình file **server.conf** trước tiên ta cần tạo 1 file theo mẫu của OpenVPN:

 # gunzip -c /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz | sudo tee /etc/openvpn/server.conf

Điều chỉnh cấu hình OpenVPN:

 # vim /etc/openvpn/server.conf

Và cấu hình như sau:

```
# Port Number.
port 1194

# TCP or UDP server.
proto udp

# Interface type, TUN or TAP.
dev tun

# Certificates.
ca ca.crt
cert server.crt
key server.key  # This file should be kept secret

cipher AES-128-CBC
auth SHA256

# Diffie hellman parameters.
dh dh1024.pem

# Subnet to use for OpenVPN Connections.
Server 10.8.0.0 255.255.255.0

#TLS auth
tls-auth ta.key 0 # This file is secret
key-direction 0

push "redirect-gateway def1 bypass-dhcp"

push "dhcp-option DNS 208.67.222.222"
push "dhcp-option DNS 208.67.220.220"


# Keepalive: send ping every 10 seconds, tunnel down after 120 seconds no response.
keepalive 10 120

# LZO Compression for the tunnel.
```

```
comp-lzo

# Drop privileges to user/group nobody.
user nobody
group nogroup

# Makes the link more resistant to connection failures.
persist-key
persist-tun

# OpenVPN Status Log files.
status openvpn-status.log

# LOG FILE VERBOSITY:
# 0 is silent, except for fatal errors
# 4 is reasonable for general usage
# 5 and 6 can help to debug connection problems
# 9 is extremely verbose
verb 3
```

Điều chỉnh cấu hình mạng trên Server

Allow IP Forwarding

 # vim /etc/sysctl.conf

tìm đến mục set net.ipv4.ip_forward uncomment và set bằng 1

 # net.ipv4.ip_forward=1

cấu hình firewall cho phép traffic từ OpenVPN Client đi qua cạc mạng

Sau khi cấu hình như trên Server OpenVPN đã sẵn sàng bạn có thể chạy dịch vụ OpenVPN và check Status

 # systemctl start openvpn@server

 # systemctl status openvpn@server

```
● openvpn@server.service - OpenVPN connection to server
   Loaded: loaded (/lib/systemd/system/openvpn@.service; disabled; vendor preset: enabled)
   Active: active (running) since T4 2017-08-02 23:16:32 ICT; 10s ago
     Docs: man:openvpn(8)
```

https://community.openvpn.net/openvpn/wiki/Openvpn23ManPage
https://community.openvpn.net/openvpn/wiki/HOWTO
Process: 1653 ExecStart=/usr/sbin/openvpn --daemon ovpn-%i --status /run/openvpn/%i.status 10 --cd /etc/openvpn --script-security 2 --config
Main PID: 1656 (openvpn)
CGroup: /system.slice/system-openvpn.slice/openvpn@server.service
├─1656 /usr/sbin/openvpn --daemon ovpn-server --status /run/openvpn/server.status 10 --cd /etc/openvpn --script-security 2 --config
└─1657 /usr/sbin/openvpn --daemon ovpn-server --status /run/openvpn/server.status 10 --cd /etc/openvpn --script-security 2 --config

Th08 02 23:16:32 ubuntu ovpn-server[1656]: /sbin/ip addr add dev tun0 local 10.8.0.1 peer 10.8.0.2
Th08 02 23:16:32 ubuntu ovpn-server[1656]: /sbin/ip route add 10.8.0.0/24 via 10.8.0.2
Th08 02 23:16:32 ubuntu ovpn-server[1656]: GID set to nogroup
Th08 02 23:16:32 ubuntu ovpn-server[1656]: UID set to nobody
Th08 02 23:16:32 ubuntu ovpn-server[1656]: UDPv4 link local (bound): [undef]
Th08 02 23:16:32 ubuntu ovpn-server[1656]: UDPv4 link remote: [undef]
Th08 02 23:16:32 ubuntu ovpn-server[1656]: MULTI: multi_init called, r=256 v=256
Th08 02 23:16:32 ubuntu ovpn-server[1656]: IFCONFIG POOL: base=10.8.0.4 size=62, ipv6=0
Th08 02 23:16:32 ubuntu ovpn-server[1656]: IFCONFIG POOL LIST
Th08 02 23:16:32 ubuntu ovpn-server[1656]: Initialization Sequence Completed

Lúc này trên OpenVPN server sẽ tạo ra một giao diện đường hầm:

 # ifconfig tun0

tun0 Link encap:UNSPEC HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
inet addr:10.8.0.1 P-t-P:10.8.0.2 Mask:255.255.255.255
UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:100

RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

***) Cấu hình phía client**

- Đầu tiên chúng ta sẽ phải cài đặt OpenVPN trên client:

 # sudo apt-get install openvpn

- Coppy tập tin certificates **ca.crt** từ server đến client bằng câu lệnh sau:

 # sftp hannv@10.0.2.15:/etc/openvpn/ca.crt /etc/openvpn/

- Cuối cùng thực hiện câu lệnh sau để kiểm tra kết quả:

 # sudo openvpn --remote 10.0.2.15 --comp-lzo --dev tun --auth-user-pass --ca ca.crt –client

- Server sẽ yêu cầu client nhập vào username và mật khẩu để xác thực, nếu thành công ta sẽ thấy như sau:

root@ubuntu:/etc/openvpn# openvpn --remote 10.0.2.15 --comp-lzo --dev tun --auth-user-pass --ca ca.crt --client

Wed Aug 2 23:44:03 2017 OpenVPN 2.3.10 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [EPOLL] [PKCS11] [MH] [IPv6] built on Jun 22 2017

Wed Aug 2 23:44:03 2017 library versions: OpenSSL 1.0.2g 1 Mar 2016, LZO 2.08

Enter Auth Username: *******

Enter Auth Password: *******

Wed Aug 2 23:44:07 2017 WARNING: No server certificate verification method has been enabled. See http://openvpn.net/howto.html#mitm for more info.

Wed Aug 2 23:44:07 2017 UDPv4 link local (bound): [undef]

Wed Aug 2 23:44:07 2017 UDPv4 link remote: [AF_INET]10.0.2.15:1194

Wed Aug 2 23:44:07 2017 WARNING: this configuration may cache passwords in memory -- use the auth-nocache option to prevent this

Wed Aug 2 23:44:07 2017 WARNING: this cipher's block size is less than 128 bit (64 bit). Consider using a --cipher with a larger block size.

Wed Aug 2 23:44:07 2017 WARNING: this cipher's block size is less than 128 bit (64 bit). Consider using a --cipher with a larger block size.

Wed Aug 2 23:44:07 2017 [server] Peer Connection Initiated with [AF_INET]10.0.2.15:1194

Wed Aug 2 23:44:10 2017 TUN/TAP device tun0 opened

Wed Aug 2 23:44:10 2017 do_ifconfig, tt->ipv6=0, tt->did_ifconfig_ipv6_setup=0

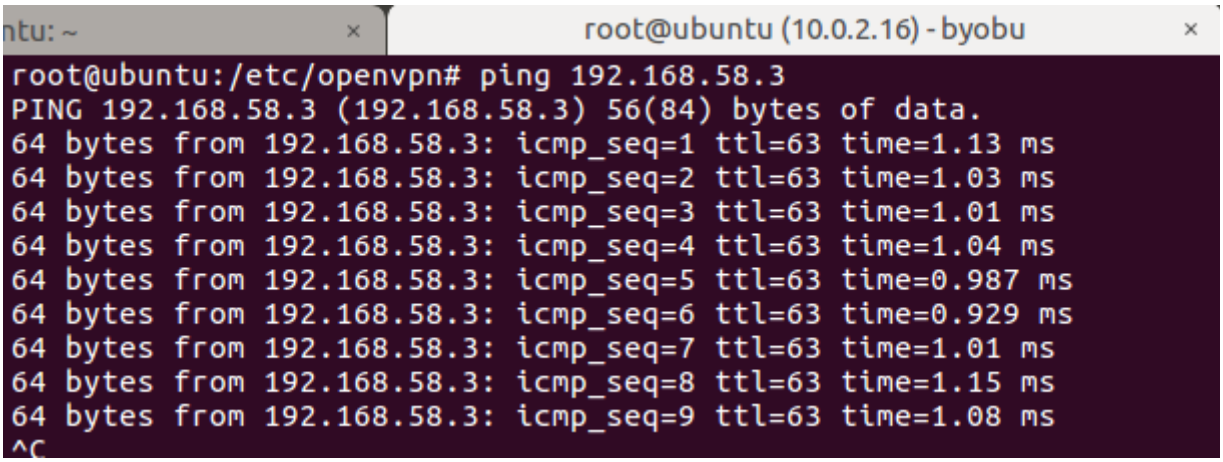
Wed Aug 2 23:44:10 2017 /sbin/ip link set dev tun0 up mtu 1500

Wed Aug 2 23:44:10 2017 /sbin/ip addr add dev tun0 local 10.8.0.10 peer 10.8.0.9

Wed Aug 2 23:44:10 2017 Initialization Sequence Completed

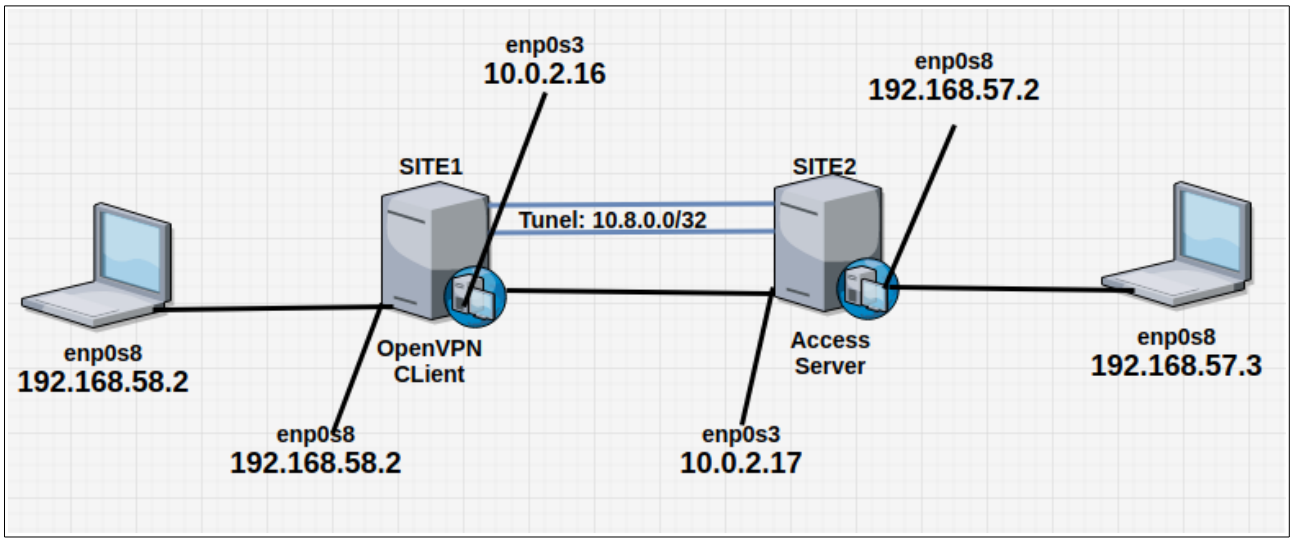
Và kết quả ta có thể thấy máy clientVPN từ 1 giải mạng khác thông qua tunel của OpenVPN và cơ chế NAT trên Server OpenVPN có thể kết nối an toàn tới các máy tính trong cùng mạng với Server

Tại máy **clientvpn :10.0.2.16** ping đến máy **Local Computer : 192.168.58.3**



II) Thiết lập OpenVPN Site to Site

1) Mô hình triển khai:



2) Cấu hình bên Access Server

Bước 1) Trước tiên tiến hành cài đặt OpenVPN

#sudo apt-get update

#sudo apt-get install openvpn easy-rsa

Bước 2) Cài đặt CA Directory

#make-cadir /etc/openvpn/openvpn-ca

- Di chuyển tới thư mục vừa khởi tạo sử dụng các câu lệnh sau

#vim vars

Bước 3) Cấu hình thông tin để tạo Certificate

export KEY_COUNTRY="VN"

export KEY_PROVINCE="HN"

export KEY_CITY="HN"

export KEY_ORG="VCCLOUD"

export KEY_EMAIL="me@myhost.mydomain"

export KEY_OU="VCCORP"

Chỉnh sửa Key_name

export KEY_NAME="server"

Bước 4) Build CA

#cd /etc/openvpn/openvpn-ca

#source vars

Sẽ có một thông báo như sau

Output

NOTE: If you run ./clean-all, I will be doing a rm -rf on /home/hannv/openvpn-ca/keys

#./clean-all

Bây giờ ta sẽ tạo root CA bằng câu lệnh sau:

#./build-ca

Generating a 2048 bit RSA private key

.....+++

.....+++

writing new private key to 'ca.key'

You are about to be asked to enter information that will be incorporated

into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

Country Name (2 letter code) [VN]:

State or Province Name (full name) [VN]:

Locality Name (eg, city) [HN City]:

Organization Name (eg, company) [VCCLOUD]:

Organizational Unit Name (eg, section) [VCCORP]:

Common Name (eg, your name or your server's hostname) [VCCLOUD CA]:

Name [server]:

Email Address [admin@email.com]:

Bước 5) Sau đó chúng ta sẽ tiến hành tạo server keys,và đặt tên là “server” các bước như sau:

#./build-key-server server

Generating a 1024 bit RSA private key

```
.....++++++
.............................++++++
writing new private key to 'server.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [VN]:
State or Province Name (full name) [HN]:
Locality Name (eg, city) [HN]:
Organization Name (eg, company) [VCCORP]:
Organizational Unit Name (eg, section) [VCCLOUD]:
Common Name (eg, your name or your server's hostname) [server]:
Name [changeme]:
Email Address [mail@host.domain]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from /etc/openvpn/openvpn-ca/server.conf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'VN'
stateOrProvinceName   :PRINTABLE:'HN'
localityName          :PRINTABLE:'HN'
organizationName      :PRINTABLE:'VCCORP'
organizationalUnitName:PRINTABLE:'changeme'
commonName            :PRINTABLE:'server'
name                  :PRINTABLE:'changeme'
emailAddress          :IA5STRING:'mail@host.domain'

Certificate is to be certified until Feb  3 09:26:00 2024 GMT (3650 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
```

```
Data Base Updated
```

Tiếp theo ta cần tạo ra một keys Diffe-Hellman đủ mạnh để sử dụng trong quá trình trao đổi khóa.

#./build-dh

```
Generating DH parameters, 1024 bit long safe prime, generator 2
This is going to take a long time
.....................................................................+.....................+.....
+.....................+.....................................
```

Sau khi tạo keys Diffe-Hellman ta cần tạo chữ ký HMAC dùng để kiểm tra tính toàn vẹn của dữ liệu khi truyền trên đường hầm OpenVPN:

 # openvpn --genkey --secret keys/ta.key

Bước 6) Tạo 1 Certificate client và Key pair cho Client

Mặc dù bước này ta có thể làm trên máy client và sau đó được xác nhận bởi server/CA nhằm mục đích đảm bảo an toàn, những trong ví dụ này tôi sẽ tạo ra các khóa đã được ký luôn trên máy chủ. Bằng câu lệnh sau:

 # cd /etc/openvpn/openvpn-ca

 # source vars

#./build-key client1

Bước 7) Cấu hình dịch vụ OpenVPN

Sao chép các tập tin cần thiết tới thư mục openvpn

 # cd /etc/openvpn/openvpn-ca/keys

 # cp ca.crt ca.key server.crt server.key ta.key dh2048.pem /etc/openvpn

Sau khi chuyển các certificates, keys đến thư mục /etc/openvpn ta tiến hành cấu hình file **server.conf** trước tiên ta cần tạo 1 file theo mẫu của OpenVPN:

 # gunzip -c /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz | sudo tee /etc/openvpn/server.conf

Điều chỉnh cấu hình OpenVPN:

 # vim /etc/openvpn/server.conf

Và cấu hình như sau:

```
# Port Number.
port 1194

# TCP or UDP server.
proto udp

# Interface type, TUN or TAP.
dev tun

# Certificates.
ca ca.crt
cert server.crt
key server.key  # This file should be kept secret

# Diffie hellman parameters.
dh dh1024.pem

# Subnet to use for OpenVPN Connections.
server 10.8.0.0 255.255.255.0

# Keepalive: send ping every 10 seconds, tunnel down after 120 seconds no response.
keepalive 10 120

# LZO Compression for the tunnel.
comp-lzo

# Drop privileges to user/group nobody.
user nobody
group nogroup

# Makes the link more resistant to connection failures.
```

```
persist-key
persist-tun

# Username and Password authentication.
client-cert-not-required
plugin /usr/lib/openvpn/openvpn-auth-pam.so login

# OpenVPN Status Log files.
status openvpn-status.log

# LOG FILE VERBOSITY:
# 0 is silent, except for fatal errors
# 4 is reasonable for general usage
# 5 and 6 can help to debug connection problems
# 9 is extremely verbose
verb 3
```

Sau khi cấu hình như trên Server OpenVPN đã sẵn sàng bạn có thể chạy dịch vụ OpenVPN và check Status

 # systemctl start openvpn@server

 # systemctl status openvpn@server

```
● openvpn@server.service - OpenVPN connection to server
   Loaded: loaded (/lib/systemd/system/openvpn@.service; disabled; vendor preset: enabled)
   Active: active (running) since T4 2017-08-02 23:16:32 ICT; 10s ago
     Docs: man:openvpn(8)
           https://community.openvpn.net/openvpn/wiki/Openvpn23ManPage
           https://community.openvpn.net/openvpn/wiki/HOWTO
  Process: 1653 ExecStart=/usr/sbin/openvpn --daemon ovpn-%i --status /run/openvpn/%i.status 10 --cd /etc/openvpn --script-security 2 --config
 Main PID: 1656 (openvpn)
   CGroup: /system.slice/system-openvpn.slice/openvpn@server.service
           ├─1656 /usr/sbin/openvpn --daemon ovpn-server --status /run/openvpn/server.status 10 --cd /etc/openvpn --script-security 2 --config
           └─1657 /usr/sbin/openvpn --daemon ovpn-server --status /run/openvpn/server.status 10 --cd /etc/openvpn --script-security 2 --config
```

Th08 02 23:16:32 ubuntu ovpn-server[1656]: /sbin/ip addr add dev tun0 local 10.8.0.1 peer 10.8.0.2

Th08 02 23:16:32 ubuntu ovpn-server[1656]: /sbin/ip route add 10.8.0.0/24 via 10.8.0.2

Th08 02 23:16:32 ubuntu ovpn-server[1656]: GID set to nogroup

Th08 02 23:16:32 ubuntu ovpn-server[1656]: UID set to nobody

Th08 02 23:16:32 ubuntu ovpn-server[1656]: UDPv4 link local (bound): [undef]

Th08 02 23:16:32 ubuntu ovpn-server[1656]: UDPv4 link remote: [undef]

Th08 02 23:16:32 ubuntu ovpn-server[1656]: MULTI: multi_init called, r=256 v=256

Th08 02 23:16:32 ubuntu ovpn-server[1656]: IFCONFIG POOL: base=10.8.0.4 size=62, ipv6=0

Th08 02 23:16:32 ubuntu ovpn-server[1656]: IFCONFIG POOL LIST

Th08 02 23:16:32 ubuntu ovpn-server[1656]: Initialization Sequence Completed

Lúc này trên OpenVPN server sẽ tạo ra một giao diện đường hầm:

 # ifconfig tun0

```
tun0   Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
       inet addr:10.8.0.1  P-t-P:10.8.0.2  Mask:255.255.255.255
       UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
       RX packets:0 errors:0 dropped:0 overruns:0 frame:0
       TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:100
       RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Bước 8) Cấu hình cho phía client trên server

Thực tế thì bước này ta có thể thực hiện tại máy client và client sẽ phải gửi các certificates, keys.. cho server để server xác thực client nhưng trong bài hướng dẫn này tôi sẽ tạo một file cấu hình trên chính Server OpenVPN và biên dịch cấu hình đó ngay trên máy server rồi mới gửi cấu hình về để client chỉ việc kích hoạt OpenVPN client và kết nối tới Server.

Trước tiên ta tạo một thư mục để lưu các tập tin cấu hình của client:

 # mkdir -p /etc/openvpn/client-configs/

Tạo một tập tin cấu hình cho client theo mẫu của OpenVPN cung cấp:

 # cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf /etc/openvpn/client-configs

Mở tập tin vừa tạo để chỉnh cấu hình lại các mục sau:

```
# The hostname/IP and port of the server.
# You can have multiple remote entries
# to load balance between the servers.
Remote 10.0.2.17 1194
proto udp
# Downgrade privileges after initialization (non-Windows only)
user nobody
group nogroup
# SSL/TLS parms.
#Uncomment các mục sau
#ca ca.crt
#cert client.crt
#key client.key
cipher AES-128-CBC
auth SHA256
key-direction 1
```

Tạo một Script tự động tạo ra cấu hình

 # vim /etc/openvpn/client-configs/make_config.sh

```
#!/bin/bash
# First argument: Client identifier
KEY_DIR=/etc/openvpn/openvpn-ca/keys
OUTPUT_DIR=/etc/openvpn/client-configs/files
BASE_CONFIG=/etc/openvpn/client-configs/base.conf
```

```
cat ${BASE_CONFIG} \
    <(echo -e '<ca>') \
    ${KEY_DIR}/ca.crt \
    <(echo -e '</ca>\n<cert>') \
    ${KEY_DIR}/${1}.crt \
    <(echo -e '</cert>\n<key>') \
    ${KEY_DIR}/${1}.key \
    <(echo -e '</key>\n<tls-auth>') \
    ${KEY_DIR}/ta.key \
    <(echo -e '</tls-auth>') \
    > ${OUTPUT_DIR}/${1}.ovpn
```

Bước 9) Generate client configurations

#cd /etc/openvpn/client-configs/

 # ./make_config.sh client1

Sau khi tạo file cấu hình bằng script trên ta cần chuyển cấu hình **client1.ovpn** sang thiết bị client. Lúc này ta cần đăng nhập vào máy OpenVPN client.

10.0.2.16$ # sftp hannv@10.0.2.17:/etc/openvpn/client-configs/client1.ovpn /etc/openvpn/

Bước 10) Install client configuration

10.0.2.16$ # apt-get install openvpn

Bây giờ ta đã sẵn sàng để kết nối giữa OpenVPN Client và Access Server bằng câu lệnh sau:

10.0.2.16$ # sudo openvpn --config /etc/openvpn/client.ovpn

4) Kiểm tra kết nối

Trên máy Server thực hiện câu lệnh:

 # ifconfig tun0

```
root@ubuntu:~# ifconfig tun0
tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:10.8.0.1  P-t-P:10.8.0.2  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Trên máy OpenVPN client thực hiện kiểm tra:

 # ifconfig tun0

```
root@ubuntu:/etc/openvpn# ifconfig tun0
tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:10.8.0.6  P-t-P:10.8.0.5  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Chú ý cấu hình IPTABLES trên máy Server và Client OpenVPN để có thể NAT các gói tin từ 2 site vào LAN của mình cấu hình như sau:

#vim /etc/ufw/before.rules

```
# rules.before
#
# Rules that should be run before the ufw command line added rules. Custom
# rules should be added to one of these chains:
#   ufw-before-input
#   ufw-before-output
#   ufw-before-forward
#
# START OPENVPN RULES
# NAT table rules
*nat
:POSTROUTING ACCEPT [0:0]
# Allow traffic from OpenVPN client to wlp11s0 (change to the interface you discovered!)
-A POSTROUTING -s 10.8.0.0/8 -o enp0s3 -j MASQUERADE
COMMIT
# END OPENVPN RULES
# Don't delete these required lines, otherwise there will be errors
*filter
```

#vim /etc/default/ufw

DEFAULT_FORWARD_POLICY="ACCEPT"