

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
CƠ SỞ TẠI THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN II



BÁO CÁO ĐỒ ÁN

Môn: An Toàn Mạng

Đề tài: Thiết lập kết nối OpenVPN cho người dùng từ xa

Giảng viên hướng dẫn:

Trần Thị Dung

Nhóm sinh viên thực hiện:

Họ tên:

MSSV:

Lê Hà Bảo Long (Nhóm Trưởng)

N18DCAT043

Lê Hà Bảo Trọng

N18DCAT095

Võ Thị Hoa Tranh

N18DCAT091

Tháng 9/2021

1. TỔNG QUAN

1.1. Khái niệm OpenVPN

OpenVPN là một phần mềm mạng riêng ảo mã nguồn mở dành cho việc tạo các đường ống (tunnel) điểm-tới-điểm được mã hóa giữa các máy chủ. Phần mềm này do James Yonan viết và được phổ biến dưới giấy phép GNU GPL.

OpenVPN cho phép các máy đồng đẳng xác thực lẫn nhau bằng một khóa bí mật được chia sẻ từ trước, chứng chỉ mã công khai (public key certificate), hoặc tên người dùng/mật khẩu. Phần mềm này được cung cấp kèm theo các hệ điều hành Solaris, Linux, OpenBSD, FreeBSD, NetBSD, Mac OS X, và Windows 2000/XP. Nó có nhiều tính năng bảo mật và kiểm soát. Nó không phải một mạng riêng ảo web, và không tương thích với IPsec hay các gói VPN khác. Toàn bộ phần mềm gồm có một file nhị phân cho cả các kết nối client và server, một file cấu hình không bắt buộc, và một hoặc nhiều file khóa tùy theo phương thức xác thực được sử dụng.



Hình 1. Logo OpenVPN

1.2. Thành phần OpenVPN

Mặc dù là giao thức mã hóa bảo mật nhất, nhưng OpenVPN vẫn dựa vào một số yếu tố quan trọng nhất định, và trừ khi VPN nhận được mọi thành phần quan trọng của giao thức, nếu không, tính bảo mật của toàn bộ giao thức mã hóa sẽ bị ảnh hưởng. Các thành phần này như sau:

- **Mật mã:** Mật mã là thuật toán mà VPN sử dụng để mã hóa dữ liệu. Khả năng mã hóa chỉ mạnh bằng mật mã mà giao thức VPN sử dụng. Các mật mã phổ biến nhất mà các nhà cung cấp VPN sử dụng là AES và Blowfish.
- **Các kênh mã hóa:** OpenVPN sử dụng hai kênh là kênh dữ liệu và kênh điều khiển. Các thành phần cho mỗi kênh như sau:
 - Kênh dữ liệu = Mật mã + Xác thực hash.

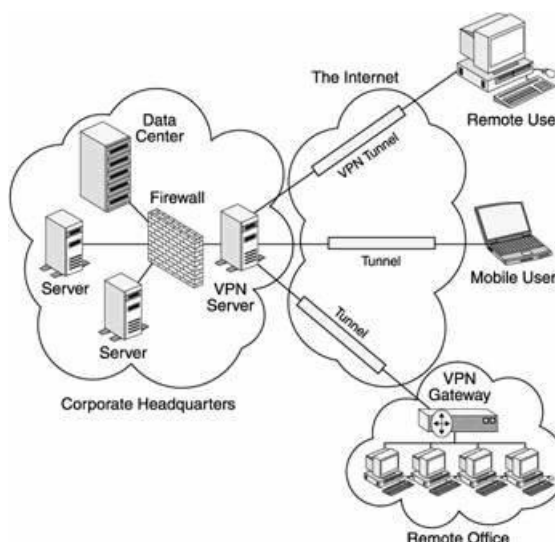
- Kênh điều khiển = Mật mã + Mã hóa handshake TLS + xác thực hash + việc Perfect Forward Secrecy có được sử dụng hay không (và được dùng như thế nào).
- **Mã hóa handshake:** Điều này được sử dụng để bảo mật trao đổi key TLS. RSA thường được sử dụng, nhưng DHE hoặc ECDH có thể được dùng thay thế và cũng cung cấp PFS.
- **Xác thực hash:** Điều này sử dụng một hàm hash mật mã để xác minh rằng dữ liệu không bị giả mạo. Trong OpenVPN, nó thường được thực hiện bằng HMAC SHA, nhưng nếu mật mã AES-GCM đang được sử dụng (thay vì AES-CBC) thì GCM có thể cung cấp xác thực hash thay thế.
- **Perfect Forward Secrecy:** PFS là một hệ thống trong đó một key mã hóa riêng tư duy nhất được tạo cho mỗi phiên. Có nghĩa là mỗi phiên Transport Layer Security (TLS) có một bộ key riêng. Chúng chỉ được sử dụng một lần và sau đó biến mất.

Những cài đặt tối thiểu được đề xuất cho các kết nối OpenVPN là:

- **Kênh dữ liệu:** Mật mã AES-128-CBC với HMAC SHA1 có xác thực. Nếu sử dụng mật mã AES-GCM thì không cần xác thực bổ sung.
- **Kênh điều khiển:** Mật mã AES-128-CBC với mã hóa handshake RSA-2048 hoặc ECDH-385 và xác thực hash HMAC SHA1. Bất kỳ quá trình trao đổi key DHE hoặc ECDH nào cũng có thể cung cấp Perfect Forward Secrecy.

1.3. Cách hoạt động OpenVPN

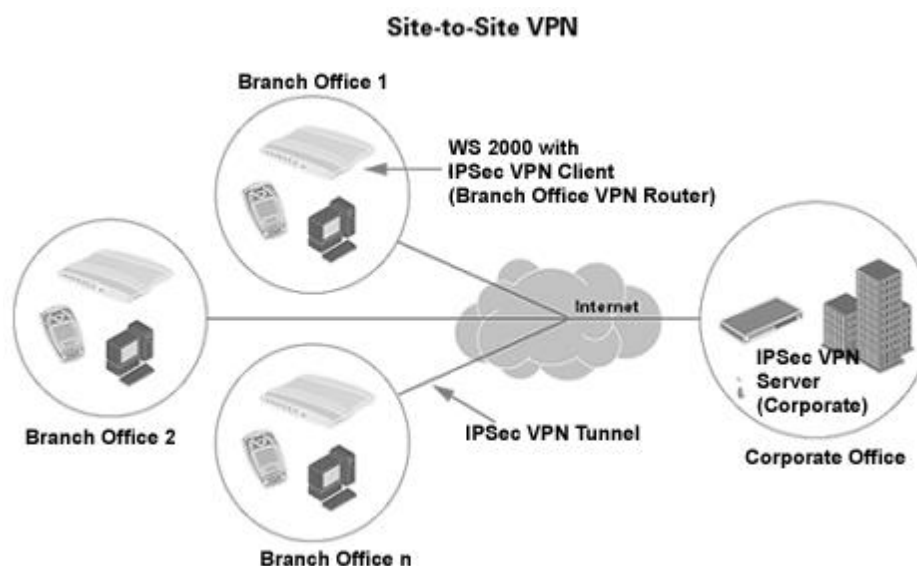
1.3.1. Cách hoạt động Remote Access VPN



Hình 2. Remote Access VPN

- Máy VPN cần kết nối (VPN Client) tạo kết nối VPN tới Server cung cấp dịch vụ VPN (VPN Server) thông qua kết nối Internet.
- Máy chủ cung cấp dịch vụ VPN trả lời kết nối tới Client
- Client gửi Certificate kết nối dựa trên file cấu hình tới Server
- Server chứng thực kết nối và cấp phép cho kết nối tới Client
- Bắt đầu trao đổi dữ liệu giữa client và server:
 - o Khi Client bắt đầu gửi dữ liệu, dữ liệu sẽ được mã hóa dựa trên khóa có trong certificate ở phía client
 - o Dữ liệu được truyền đến Server
 - o Sau khi Server nhận được dữ liệu, sẽ tiến hành giải mã dữ liệu dựa trên khóa trên Server

1.3.2. Site-to-Site (Lan-to-Lan)



Hình 3. Site to Site VPN

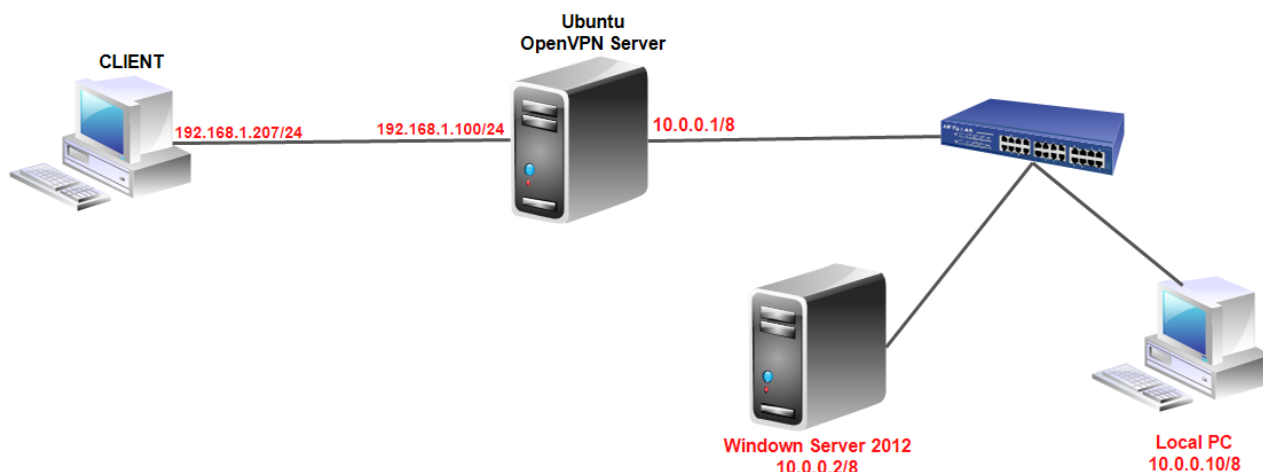
Với OpenVPN Access Server, VPN site-to-site đơn giản như kết nối bộ định tuyến. Trước tiên, Access Server sẽ được thiết lập tại trụ sở chính, sau đó mỗi vị trí bổ sung được thiết lập với một bộ định tuyến có cấu hình kết nối người dùng. Đảm bảo sử dụng bộ định tuyến chạy chương trình cơ sở DD-WRT mà có thể tìm thấy ở đây nếu bộ định tuyến chưa có (hoặc bất kỳ bộ định tuyến nào tương thích với ứng dụng khách OpenVPN). Sau khi cấu hình cài đặt Máy chủ truy cập cho phù hợp và thiết lập cấu hình kết nối người dùng trên bộ định tuyến của mình, người dùng đã kết nối với mạng riêng HQ của mình.

Sau đó, khi người dùng kết nối với mạng tại các địa điểm này, họ sẽ tự động được kết nối với tất cả các tài nguyên mà HQ cung cấp. Thiết lập mạng dễ dàng thiết lập, cho dù địa điểm mới là văn phòng tại nhà hay cửa hàng pop-up du lịch: tất cả những gì phải làm là bật nguồn bộ định tuyến có kết nối internet.

Điều này có nghĩa là các biện pháp kiểm soát an ninh HQ có thể được thực thi từng nơi, và các tài nguyên của mạng HQ có thể truy cập được ở tất cả các địa điểm!

2. THỰC HÀNH

2.1. Mô hình kết nối



Hình 4. Mô hình kết nối OpenVPN

STT	Server Name		Interface 1	Interface 2	Interface 3
1	Window Server 2012 (Domain Controller)	IP	10.0.0.2		
		SM	255.0.0.0		
		DG	10.0.0.1		
		DNS			
2	OpenVPN Server (Ubuntu)	IP	10.0.0.1	DHCP	
		SM	255.0.0.0		
		DG			
		DNS			
3	Local PC	IP	10.0.0.10		
		SM	255.0.0.0		
		DG	10.0.0.1		
		DNS			
4	Client PC	IP	DHCP		
		SM			
		DG			
		DNS			

2.2. Thực hiện

2.2.1. Cài đặt và cấu hình OpenVPN

• Bước 1: Cài đặt OpenVPN và Easy-RSA

Easy-RSA là công cụ quản lý cơ sở hạ tầng public key (PKI) mà em sẽ sử dụng trên Server OpenVPN để tạo certificate request.

Cài đặt OpenVPN, Easy-RSA và SSH:

```
sudo su -
```

```
apt install easy-rsa ssh openvpn
```

```
n18dcat091@ubuntu:~/Desktop$ sudo su -
[sudo] password for n18dcat091:
root@ubuntu:~# apt install easy-rsa ssh openvpn
Reading package lists... Done
Building dependency tree
Reading state information... Done
openvpn is already the newest version (2.4.7-1ubuntu2.20.04.3).
openvpn set to manually installed.
The following additional packages will be installed:
  libccid ncurses-term openssl openssl-pkcs11 openssh-server openssh-sftp-server pcscd ssh-import-id
Suggested packages:
  molly-guard monkeysphere ssh-askpass
The following NEW packages will be installed:
  easy-rsa libccid ncurses-term openssl openssl-pkcs11 openssh-server openssh-sftp-server pcscd ssh ssh-import-id
```

Hình 5. Cài đặt Easy-RSA SSH và OpenVPN

Tạo một folder mới trên Server OpenVPN: `mkdir ~/easy-rsa`

Tạo một softlink để khi có bất kỳ cập nhật nào đối với gói easy-rsa sẽ được tự động phản ánh trong các tập lệnh PKI: `ln -s /usr/share/easy-rsa/* ~/easy-rsa/`

```
Selecting previously unselected package openssl.
Preparing to unpack .../7-openssl_0.20.0-3_amd64.deb ...
Unpacking openssl (0.20.0-3) ...
Selecting previously unselected package easy-rsa.
Preparing to unpack .../8-easy-rsa_3.0.6-1_all.deb ...
Unpacking easy-rsa (3.0.6-1) ...
Selecting previously unselected package ssh-import-id.
Preparing to unpack .../9-ssh-import-id_5.10-0ubuntu1_all.deb ...
Unpacking ssh-import-id (5.10-0ubuntu1) ...
Setting up openssh-sftp-server (1:8.2p1-4ubuntu0.3) ...
Setting up openssh-server (1:8.2p1-4ubuntu0.3) ...

Creating config file /etc/ssh/sshd_config with new version
Creating SSH2 RSA key; this may take some time ...
3072 SHA256:y1pw7nBF9jJ+dHC45xP4VjCWfapeSQ7sCkJsyyWBTM root@ubuntu (RSA)
Creating SSH2 ECDSA key; this may take some time ...
256 SHA256:3D8sULFHAM0nEHAaX1zBPTB6+sf9Fnpzz2bd+hI0TSu root@ubuntu (ECDSA)
Creating SSH2 ED25519 key; this may take some time ...
256 SHA256:Ji8CqjUoFatQgyUhzgcjIbJk0E08mBqAPeX1/1DJLFo root@ubuntu (ED25519)
Created symlink /etc/systemd/system/ssh.service → /lib/systemd/system/ssh.service.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /lib/systemd/system/ssh.service.
rescue-ssh.target is a disabled or a static unit, not starting it.
Setting up libccid (1.4.31-1) ...
Setting up ssh-import-id (5.10-0ubuntu1) ...
Attempting to convert /etc/ssh/ssh_import_id
Setting up pcscd (1.8.26-3) ...
Created symlink /etc/systemd/system/sockets.target.wants/pcscd.socket → /lib/systemd/system/pcscd.socket.
pcscd.service is a disabled or a static unit, not starting it.
Setting up openssl-pkcs11:amd64 (0.20.0-3) ...
Setting up easy-rsa (3.0.6-1) ...
Setting up ncurses-term (6.2-0ubuntu2) ...
Setting up ssh (1:8.2p1-4ubuntu0.3) ...
Setting up openssl (0.20.0-3) ...
Processing triggers for ufw (0.36-6) ...
Processing triggers for systemd (245.4-4ubuntu3.13) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for desktop-file-utils (0.24-1ubuntu3) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
root@ubuntu:~# mkdir ~/easy-rsa
root@ubuntu:~# ln -s /usr/share/easy-rsa/* ~/easy-rsa/
```

Hình 6. Tạo Folder ~/easy-rsa

• Bước 2: Tạo PKI

Tạo folder để quản lý các certificate request của server và client.

cd vào folder easy-rsa, tạo và chỉnh sửa file vars.


```
cd ~/easy-rsa
```

```
nano vars
```

```
GNU nano 4.8 vars
set_var EASYRSA_REQ_COUNTRY "VN"
set_var EASYRSA_REQ_PROVINCE "Thu Duc"
set_var EASYRSA_REQ_CITY "Ho Chi Minh City"
set_var EASYRSA_REQ_ORG "OpenVPN"
set_var EASYRSA_REQ_EMAIL "nhom6ATM.local"
set_var EASYRSA_REQ_OU "n18dcat091"
set_var EASYRSA_ALGO "ec"
set_var EASYRSA_DIGEST "sha512"
```

Hình 7. Tạo và chỉnh sửa file vars

Khi đã điền file vars, em tiếp tục tạo folder PKI: `./easyrsa init-pki`

```
root@ubuntu:~/easy-rsa# ./easyrsa init-pki
Note: using Easy-RSA configuration from: ./vars
init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /root/easy-rsa/pki
```

Hình 8. Tạo folder PKI

• Bước 3: Tạo Yêu cầu Chứng chỉ Server OpenVPN và Private Key

Tạo cặp khóa public key và private key root cho CA: `./easyrsa build-ca nopass`

```
root@ubuntu:~/easy-rsa# ./easyrsa build-ca nopass
Note: using Easy-RSA configuration from: ./vars
Using SSL: openssl OpenSSL 1.1.1f 31 Mar 2020
read EC key
writing EC key
Can't load /root/easy-rsa/pki/.rnd into RNG
140212507952448:error:2406F079:random number generator:RAND_load_file:Cannot open file:../crypto/rand/randfile.c:
98:Filename=/root/easy-rsa/pki/.rnd
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.
-----
Common Name (eg: your user, host, or server name) [Easy-RSA CA]:
CA creation complete and you may now import and sign cert requests.
Your new CA certificate file for publishing is at:
/root/easy-rsa/pki/ca.crt
```

Hình 9. Tạo cặp khóa public key và private key root cho CA

```
cp ~/easy-rsa/pki/ca.crt /usr/local/share/ca-certificates/
```

```
update-ca-certificates
```


Gọi easysrsa với tùy chọn gen-req. Trong cấu hình này, tên của Server OpenVPN sẽ là server, bao gồm tùy chọn nopass: *./easysrsa gen-req server nopass*

```
root@ubuntu:~/easy-rsa# cp ~/easy-rsa/pki/ca.crt /usr/local/share/ca-certificates/
root@ubuntu:~/easy-rsa# update-ca-certificates
Updating certificates in /etc/ssl/certs...
1 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
root@ubuntu:~/easy-rsa# ./easysrsa gen-req server nopass

Note: using Easy-RSA configuration from: ./vars

Using SSL: openssl OpenSSL 1.1.1f  31 Mar 2020
Generating an EC private key
writing new private key to '/root/easy-rsa/pki/private/server.key.cpYlPvh6TH'
-----
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.
-----
Common Name (eg: your user, host, or server name) [server]:

Keypair and certificate request completed. Your files are:
req: /root/easy-rsa/pki/reqs/server.req
key: /root/easy-rsa/pki/private/server.key
```

Hình 10. Tạo private key cho server và file certificate request được gọi là server.req.

- **Bước 4: Ký Yêu cầu Chứng chỉ của Server OpenVPN**

Ký yêu cầu bằng cách chạy tập lệnh easysrsa với tùy chọn yêu cầu sign-req. Vì đang làm việc với certificate request của server OpenVPN, nên em sử dụng loại yêu cầu server.

./easysrsa sign-req server server

```
root@ubuntu:~/easy-rsa# ./easyrsa sign-req server server
Note: using Easy-RSA configuration from: ./vars
Using SSL: openssl OpenSSL 1.1.1f 31 Mar 2020

You are about to sign the following certificate.
Please check over the details shown below for accuracy. Note that this request
has not been cryptographically verified. Please be sure it came from a trusted
source or that you have verified the request checksum with the sender.

Request subject, to be signed as a server certificate for 1080 days:

subject=
  commonName                = server

Type the word 'yes' to continue, or any other input to abort.
  Confirm request details: yes
Using configuration from /root/easy-rsa/pki/safessl-easyrsa.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName      :ASN.1 12:'server'
Certificate is to be certified until Sep 20 21:11:20 2024 GMT (1080 days)

Write out database with 1 new entries
Data Base Updated

Certificate created at: /root/easy-rsa/pki/issued/server.crt
```

Hình 11. Ký Yêu cầu Chứng chỉ của Server OpenVPN

Bây giờ, sao chép các file sang /etc/openvpn/server

```
cp ~/easy-rsa/pki/private/server.key /etc/openvpn/server/
```

```
cp ~/easy-rsa/pki/issued/server.crt /etc/openvpn/server/
```

```
cp ~/easy-rsa/pki/ca.crt /etc/openvpn/server/
```

• Bước 5: Cấu hình tài liệu mật mã OpenVPN

Đối với một lớp bảo mật bổ sung, em sẽ thêm một key bí mật được chia sẻ bổ sung mà server và tất cả các client sẽ sử dụng với chỉ thị `tls-crypt` của OpenVPN. Tùy chọn này được sử dụng để làm xáo trộn certificate TLS được sử dụng khi server và client kết nối với nhau lần đầu. Nó cũng được sử dụng bởi server OpenVPN để thực hiện kiểm tra nhanh các gói đến: nếu một gói được ký bằng khóa chia sẻ trước, thì server sẽ xử lý nó; nếu nó không được ký, thì server biết nó đến từ một nguồn không tin cậy và có thể loại bỏ nó.

Tùy chọn này sẽ giúp đảm bảo server OpenVPN có thể đối phó với lưu lượng truy cập không được xác thực, quét cổng và các cuộc tấn công từ chối dịch vụ, có thể làm mất tài nguyên server. Nó cũng làm cho việc xác định lưu lượng mạng OpenVPN khó hơn.

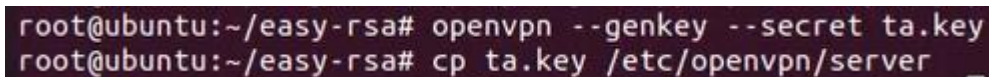
Để tạo khóa chia sẻ trước tls-crypt:

```
cd ~/easy-rsa
```

```
openvpn --genkey --secret ta.key
```

Kết quả sẽ là một file có tên ta.key. Sao chép nó vào folder /etc/openvpn/server/

```
cp ta.key /etc/openvpn/server
```



```
root@ubuntu:~/easy-rsa# openvpn --genkey --secret ta.key
root@ubuntu:~/easy-rsa# cp ta.key /etc/openvpn/server
```

Hình 12. Cấu hình tài liệu mật mã OpenVPN

• Bước 6: Tạo certificate ứng dụng client và cặp khóa

Em sẽ tạo một cặp certificate và khóa ứng dụng với cặp khóa /certificate đầu tiên được gọi là client1.

Bắt đầu bằng cách tạo folder trong folder chính để lưu trữ certificate ứng dụng client và các file khóa:

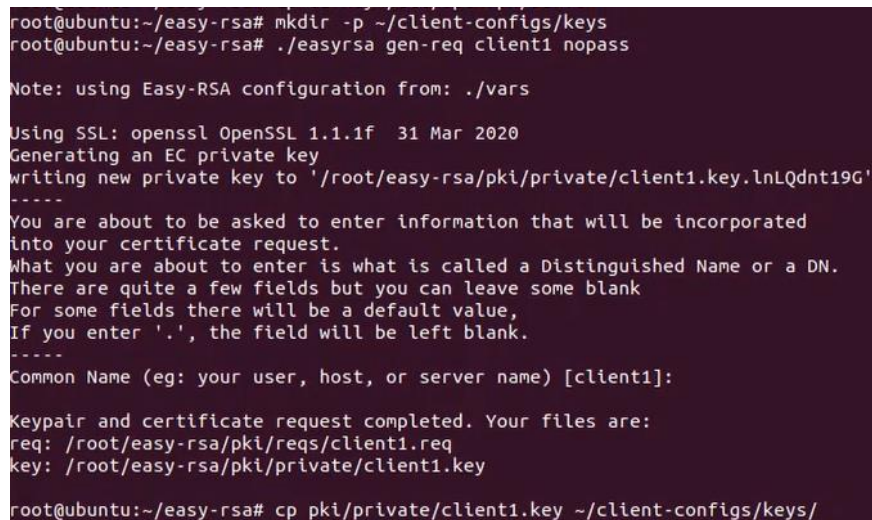
```
mkdir -p ~/client-configs/keys
```

Chạy tập lệnh easysrsa với các tùy chọn gen-req và nopass, cùng với tên chung cho ứng dụng client.

```
./easysrsa gen-req client1 nopass
```

Sao chép file client1.key vào folder ~/client-configs/keys/

```
cp pki/private/client1.key ~/client-configs/keys/
```



```
root@ubuntu:~/easy-rsa# mkdir -p ~/client-configs/keys
root@ubuntu:~/easy-rsa# ./easysrsa gen-req client1 nopass

Note: using Easy-RSA configuration from: ./vars

Using SSL: openssl OpenSSL 1.1.1f  31 Mar 2020
Generating an EC private key
writing new private key to '/root/easy-rsa/pki/private/client1.key.lnLQdnt19G'
-----
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.
-----
Common Name (eg: your user, host, or server name) [client1]:

Keypair and certificate request completed. Your files are:
req: /root/easy-rsa/pki/reqs/client1.req
key: /root/easy-rsa/pki/private/client1.key

root@ubuntu:~/easy-rsa# cp pki/private/client1.key ~/client-configs/keys/
```

Hình 13. Tạo cặp khóa cho client

Tiếp theo, ký tên vào yêu cầu giống như đã làm đối với server ở bước trước. Tuy nhiên, lần này chỉ định loại yêu cầu của client: `./easyrsa sign-req client client1`

Khi được yêu cầu, nhập yes để xác nhận.

Thao tác này sẽ tạo file certificate ứng dụng client có tên `client1.crt`.

Sao chép certificate ứng dụng client vào folder `~/client-configs/keys/`

`cp pki/issued/client1.crt ~/client-configs/keys/`

Tiếp theo, sao chép các file `ca.crt` và `ta.key` vào folder `~/client-configs/keys/`

`cp ~/easy-rsa/ta.key ~/client-configs/keys/`

`cp /etc/openvpn/server/ca.crt ~/client-configs/keys/`

```
root@ubuntu:~/easy-rsa# ./easyrsa sign-req client client1

Note: using Easy-RSA configuration from: ./vars

Using SSL: openssl OpenSSL 1.1.1f  31 Mar 2020

You are about to sign the following certificate.
Please check over the details shown below for accuracy. Note that this request
has not been cryptographically verified. Please be sure it came from a trusted
source or that you have verified the request checksum with the sender.

Request subject, to be signed as a client certificate for 1080 days:

subject=
  commonName              = client1

Type the word 'yes' to continue, or any other input to abort.
Confirm request details: yes
Using configuration from /root/easy-rsa/pki/safessl-easyrsa.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName      :ASN.1 12:'client1'
Certificate is to be certified until Sep 20 21:12:45 2024 GMT (1080 days)

Write out database with 1 new entries
Data Base Updated

Certificate created at: /root/easy-rsa/pki/issued/client1.crt

root@ubuntu:~/easy-rsa# cp pki/issued/client1.crt ~/client-configs/keys/
root@ubuntu:~/easy-rsa# cp ~/easy-rsa/ta.key ~/client-configs/keys/
root@ubuntu:~/easy-rsa# cp /etc/openvpn/server/ca.crt ~/client-configs/keys/
```

Hình 14. Tạo certificate cho client

• Bước 7: Cấu hình OpenVPN

Đầu tiên, sao chép file `server.conf` mẫu làm điểm bắt đầu cho file cấu hình

`cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz`

`/etc/openvpn/server/`

`gunzip /etc/openvpn/server/server.conf.gz`


```
root@ubuntu:~/easy-rsa# cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz /etc/openvpn/server/
root@ubuntu:~/easy-rsa# gunzip /etc/openvpn/server/server.conf.gz
```

Mở file mới để chỉnh sửa:

nano /etc/openvpn/server/server.conf

```
# Diffie hellman parameters.
# Generate your own with:
#   openssl dhparam -out dh2048.pem 2048
;dh dh2048.pem
dh none

# Push routes to the client to allow it
# to reach other private subnets behind
# the server. Remember that these
# private subnets will also need
# to know to route the OpenVPN client
# address pool (10.8.0.0/255.255.255.0)
# back to the OpenVPN server.
push "route 192.168.10.0 255.255.255.0"
;push "route 192.168.20.0 255.255.255.0"

# The server and each client must have
# a copy of this key.
# The second parameter should be '0'
# on the server and '1' on the clients.
;tls-auth ta.key 0 # This file is secret
tls-crypt ta.key

# Select a cryptographic cipher.
# This config item must be copied to
# the client config file as well.
# Note that v2.4 client/server will automatically
# negotiate AES-256-GCM in TLS mode.
# See also the ncp-cipher option in the manpage
;cipher AES-256-CBC
cipher AES-256-GCM
auth SHA256

# You can uncomment this out on
# non-Windows systems.
user nobody
group nogroup
```

Hình 15. Chỉnh sửa file server.conf

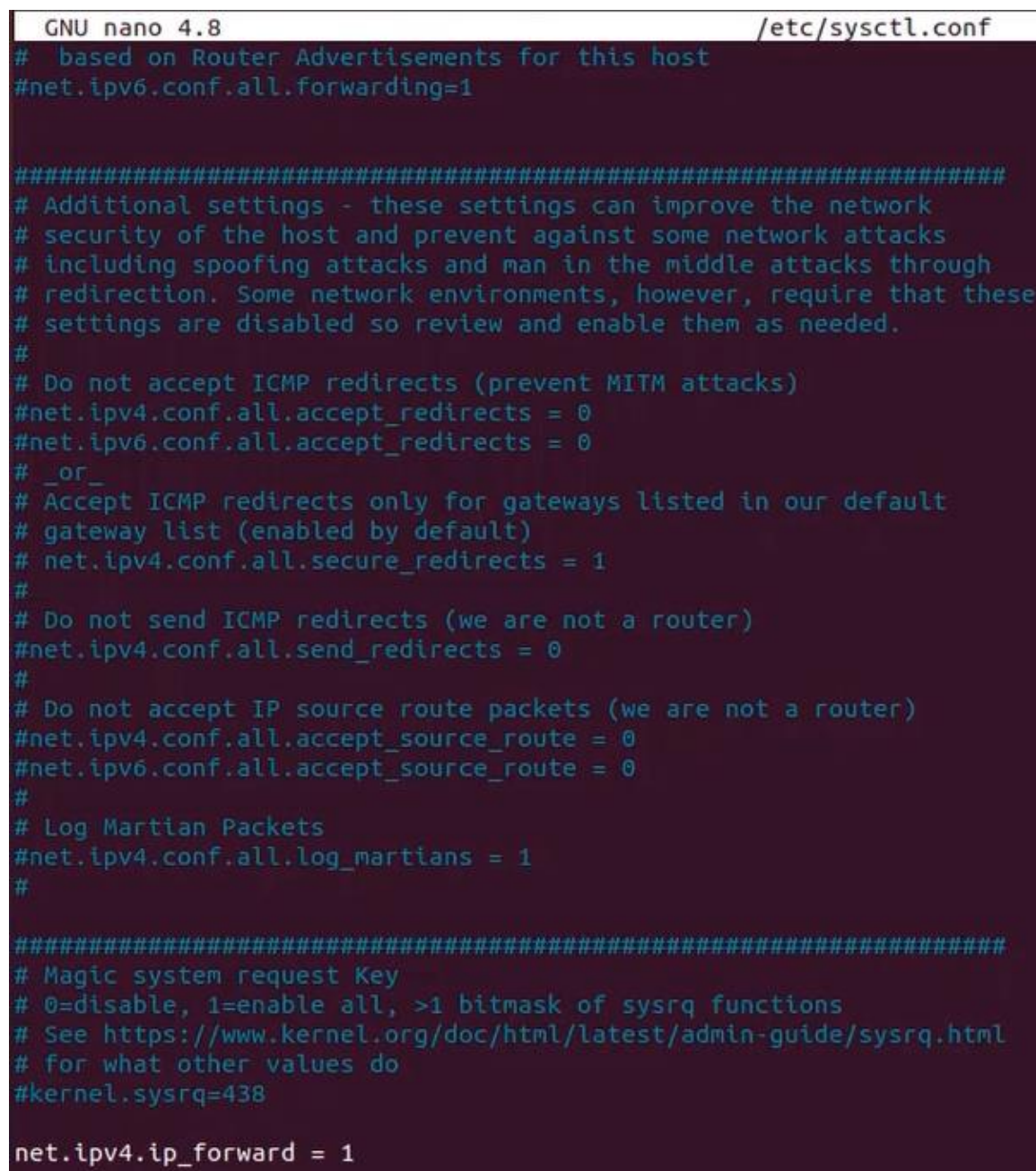
• Bước 8: Điều chỉnh cấu hình mạng server OpenVPN

Có một số khía cạnh của cấu hình mạng của server cần được tinh chỉnh để OpenVPN có thể định tuyến chính xác lưu lượng truy cập thông qua VPN. Đầu tiên trong số này là

chuyển tiếp IP, một phương pháp để xác định nơi lưu lượng truy cập IP nên được định tuyến. Đây là điều cần thiết đối với chức năng VPN mà server sẽ cung cấp.

Để điều chỉnh cài đặt chuyển tiếp IP mặc định của server OpenVPN, mở file `/etc/sysctl.conf`: `nano /etc/sysctl.conf`

Sau đó, thêm dòng sau vào cuối file: `/etc/sysctl.conf`



```
GNU nano 4.8 /etc/sysctl.conf
# based on Router Advertisements for this host
#net.ipv6.conf.all.forwarding=1

#####
# Additional settings - these settings can improve the network
# security of the host and prevent against some network attacks
# including spoofing attacks and man in the middle attacks through
# redirection. Some network environments, however, require that these
# settings are disabled so review and enable them as needed.
#
# Do not accept ICMP redirects (prevent MITM attacks)
#net.ipv4.conf.all.accept_redirects = 0
#net.ipv6.conf.all.accept_redirects = 0
# _or_
# Accept ICMP redirects only for gateways listed in our default
# gateway list (enabled by default)
# net.ipv4.conf.all.secure_redirects = 1
#
# Do not send ICMP redirects (we are not a router)
#net.ipv4.conf.all.send_redirects = 0
#
# Do not accept IP source route packets (we are not a router)
#net.ipv4.conf.all.accept_source_route = 0
#net.ipv6.conf.all.accept_source_route = 0
#
# Log Martian Packets
#net.ipv4.conf.all.log_martians = 1
#
#####
# Magic system request Key
# 0=disable, 1=enable all, >1 bitmask of sysrq functions
# See https://www.kernel.org/doc/html/latest/admin-guide/sysrq.html
# for what other values do
#kernel.sysrq=438

net.ipv4.ip_forward = 1
```

Hình 16. Chỉnh sửa file `sysctl.conf`

Bây giờ server OpenVPN có thể chuyển tiếp lưu lượng đến từ một thiết bị ethernet khác. Cấu hình này sẽ định tuyến tất cả lưu lượng truy cập web từ client thông qua địa chỉ IP của server và địa chỉ IP công cộng của client sẽ bị ẩn một cách hiệu quả.

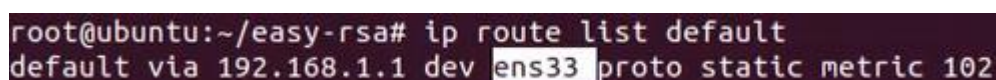
• Bước 9: Cấu hình firewall

Quy định cách server xử lý lưu lượng client bằng cách cài đặt một số luật firewall và cấu hình định tuyến.

Để cho phép OpenVPN thông qua firewall, em bật giả mạo, một khái niệm iptables cung cấp tính năng dịch địa chỉ mạng động (NAT) nhanh chóng để định tuyến chính xác các kết nối client.

Trước khi mở file cấu hình firewall để thêm các luật giả mạo, trước tiên em tìm network interface công cộng của máy mình. Để làm điều này, nhập:

ip route list default



```
root@ubuntu:~/easy-rsa# ip route list default
default via 192.168.1.1 dev ens33 proto static metric 102
```

Hình 17. Kiểm tra network interface

Mở file `/etc/ufw/before.rules` để thêm cấu hình có liên quan:

nano /etc/ufw/before.rules

Các luật UFW thường được thêm vào bằng lệnh `ufw`. Tuy nhiên, các luật được liệt kê trong file `before.rules` được đọc và đưa vào vị trí trước khi các luật UFW thông thường được tải.

/etc/ufw/before.rules



```
---
GNU nano 4.8 /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 eth0 (change to the interface you discovered!)
-A POSTROUTING -s 10.8.0.0/8 -o ens33 -j MASQUERADE
COMMIT
# END OPENVPN RULES
```

Hình 18. Chỉnh sửa file `before.rules`

Tiếp theo, yêu cầu UFW cho phép các gói được chuyển tiếp theo mặc định. Để thực hiện việc này, hãy mở file `/etc/default/ufw`

nano /etc/default/ufw

Bên trong, tìm chỉ thị `DEFAULT_FORWARD_POLICY` và thay đổi giá trị từ `DROP` thành `ACCEPT`: `DEFAULT_FORWARD_POLICY="ACCEPT"`

```
# Set the default forward policy to ACCEPT, DROP or REJECT. Please note that
# if you change this you will most likely want to adjust your rules
DEFAULT_FORWARD_POLICY="ACCEPT"
```

Hình 19.

Tiếp theo, điều chỉnh chính firewall để cho phép lưu lượng truy cập vào OpenVPN.

```
ufw allow 1194/udp
```

Sau khi thêm các luật đó, tắt và bật lại UFW để khởi động lại nó và tải các thay đổi từ tất cả các file đã sửa đổi:

```
ufw disable
```

```
ufw enable
```

```
root@ubuntu:~/easy-rsa# ufw allow 1194/udp
Rules updated
Rules updated (v6)
root@ubuntu:~/easy-rsa# ufw disable
Firewall stopped and disabled on system startup
root@ubuntu:~/easy-rsa# ufw enable
Firewall is active and enabled on system startup
```

Hình 20. Restart firewall

• Bước 10: Khởi động OpenVPN

OpenVPN chạy như một dịch vụ systemd, vì vậy em có thể sử dụng `systemctl` để quản lý nó. Em sẽ cấu hình OpenVPN để khởi động khi server khởi động, để thực hiện việc này, em bật dịch vụ OpenVPN bằng cách thêm nó vào `systemctl`

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

Sau đó khởi động dịch vụ OpenVPN:

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

Kiểm tra kỹ xem dịch vụ OpenVPN có đang hoạt động hay không bằng lệnh sau. Em thấy `active (running)` trong kết quả

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

```

root@ubuntu:~/easy-rsa# systemctl -f enable openvpn-server@server.service
Created symlink /etc/systemd/system/multi-user.target.wants/openvpn-server@server.service → /lib/systemd/system/o
penvpn-server@server.service.
root@ubuntu:~/easy-rsa# systemctl start openvpn-server@server.service
root@ubuntu:~/easy-rsa# systemctl status openvpn-server@server.service
● openvpn-server@server.service - OpenVPN service for server
   Loaded: loaded (/lib/systemd/system/openvpn-server@server.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2021-10-06 14:19:00 PDT; 8s ago
     Docs: man:openvpn(8)
           https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
           https://community.openvpn.net/openvpn/wiki/HOWTO
   Main PID: 37009 (openvpn)
   Status: "Initialization Sequence Completed"
     Tasks: 1 (limit: 2273)
    Memory: 864.0K
    CGroup: /system.slice/system-openvpn\x2dservice.slice/openvpn-server@server.service
            └─37009 /usr/sbin/openvpn --status /run/openvpn-server/status-server.log --status-version 2 --suppr
Oct 06 14:19:00 ubuntu openvpn[37009]: Could not determine IPv4/IPv6 protocol. Using AF_INET
Oct 06 14:19:00 ubuntu openvpn[37009]: Socket Buffers: R=[212992->212992] S=[212992->212992]
Oct 06 14:19:00 ubuntu openvpn[37009]: UDPv4 link local (bound): [AF_INET][undef]:1194
Oct 06 14:19:00 ubuntu openvpn[37009]: UDPv4 link remote: [AF_UNSPEC]
Oct 06 14:19:00 ubuntu openvpn[37009]: GID set to nogroup
Oct 06 14:19:00 ubuntu openvpn[37009]: UID set to nobody
Oct 06 14:19:00 ubuntu openvpn[37009]: MULTI: multi init called, r=256 v=256
Oct 06 14:19:00 ubuntu openvpn[37009]: IFCONFIG POOL: base=10.8.0.4 size=62, ipv6=0
Oct 06 14:19:00 ubuntu openvpn[37009]: IFCONFIG POOL LIST
Oct 06 14:19:00 ubuntu openvpn[37009]: Initialization Sequence Completed
lines 1-23/23 (END)

```

Hình 21. Restart service OpenVPN

• Bước 11: Tạo cơ sở hạ tầng cấu hình client

Bắt đầu bằng cách tạo một folder mới, nơi em sẽ lưu trữ các file cấu hình client trong folder cấu hình client-configs đã tạo trước đó.

```
mkdir -p ~/client-configs/files
```

Tiếp theo, sao chép file cấu hình client mẫu vào folder cấu hình client-configs để sử dụng làm cấu hình cơ sở

```
cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf ~/client-
configs/base.conf
```

```

root@ubuntu:~/easy-rsa# mkdir -p ~/client-configs/files
root@ubuntu:~/easy-rsa# cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf ~/client-
configs/base.conf

```

Hình 22. Backup file cấu hình client

Mở file mới này và chỉnh sửa

```
nano ~/client-configs/base.conf
```

```

# The hostname/IP and port of the server.
# You can have multiple remote entries
# to load balance between the servers.
remote 192.168.1.100 1194
;remote my-server-2 1194

# Downgrade privileges after initialization (non-Windows only)
user nobody
group nogroup

```

```
# SSL/TLS parms.
# See the server config file for more
# description.  It's best to use
# a separate .crt/.key file pair
# for each client.  A single ca
# file can be used for all clients.
;ca ca.crt
;cert client.crt
;key client.key

# If a tls-auth key is used on the server
# then every client must also have the key.
;tls-auth ta.key 1

# Select a cryptographic cipher.
# If the cipher option is used on the server
# then you must also specify it here.
# Note that v2.4 client/server will automatically
# negotiate AES-256-GCM in TLS mode.
# See also the ncp-cipher option in the manpage
cipher AES-256-CBC
auth SHA256

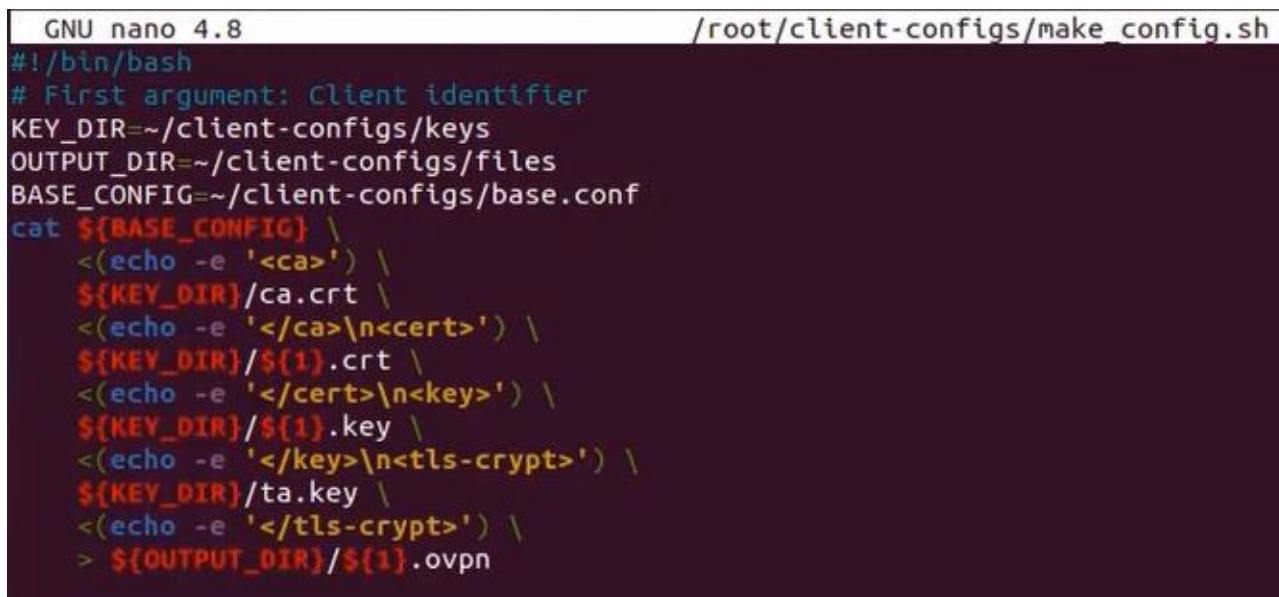
# Silence repeating messages
;mute 20
key-direction 1

; script-security 2
; up /etc/openvpn/update-resolv-conf
; down /etc/openvpn/update-resolv-conf
; script-security 2
; up /etc/openvpn/update-systemd-resolved
; down /etc/openvpn/update-systemd-resolved
; down-pre
; dhcp-option DOMAIN-ROUTE
```

Hình 23. Chỉnh sửa file cấu hình client

Tiếp theo, em sẽ tạo một tập lệnh biên dịch cấu hình cơ sở với certificate, khóa và file mã hóa có liên quan, sau đó đặt cấu hình đã tạo vào folder ~/client-configs/files.

```
nano ~/client-configs/make_config.sh
```

```
GNU nano 4.8 /root/client-configs/make_config.sh
#!/bin/bash
# First argument: Client identifier
KEY_DIR=~/.client-configs/keys
OUTPUT_DIR=~/.client-configs/files
BASE_CONFIG=~/.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-crypt>' ) \
  ${KEY_DIR}/ta.key \
  <(echo -e '</tls-crypt>' ) \
  > ${OUTPUT_DIR}/${1}.ovpn
```

Hình 24.

Trước khi tiếp tục, hãy nhớ đánh dấu file này là file thực thi bằng lệnh :

```
chmod 700 ~/.client-configs/make_config.sh
```

Tập lệnh này sẽ tạo một bản sao của file *base.conf* đã tạo, thu thập tất cả certificate và file khóa đã tạo cho ứng dụng client, extract nội dung của chúng, nối chúng vào bản sao của file cấu hình cơ sở và xuất tất cả những thứ này nội dung vào file cấu hình client mới. Điều này nghĩa là thay vì phải quản lý các file cấu hình, certificate và khóa của khách hàng một cách riêng biệt, tất cả thông tin cần thiết được lưu trữ ở một nơi. Lợi ích của việc sử dụng phương pháp này là nếu em cần thêm ứng dụng client trong tương lai, em có thể chạy tập lệnh này để nhanh chóng tạo file cấu hình mới và đảm bảo tất cả thông tin quan trọng được lưu trữ trong một file duy nhất, dễ truy cập vị trí.

• Bước 12: Tạo cấu hình client

Hiện tại em đã tạo certificate ứng dụng client và khóa có tên *client1.crt* và *client1.key*. Em sẽ tạo file cấu hình cho các thông tin đăng nhập này:

```
cd ~/.client-configs
./make_config.sh client1
```

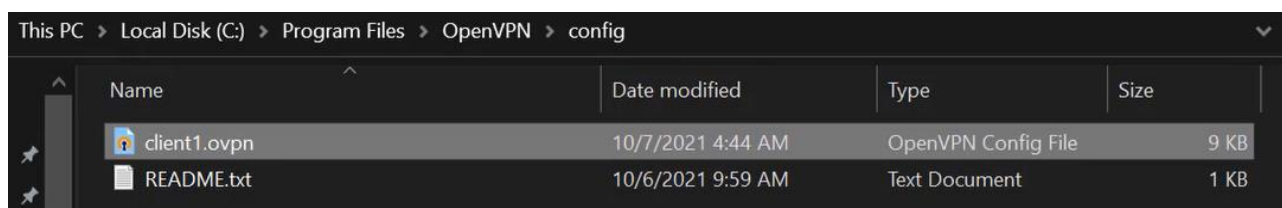
• Bước 13: Cài đặt cấu hình client

Download ứng dụng client OpenVPN dành cho Windows từ trang Download của OpenVPN. Chọn version thích hợp cho version Windows.

Lưu ý: OpenVPN cần có quyền quản trị để cài đặt.

Sau khi cài đặt OpenVPN, sao chép file *client1.ovpn* vào:

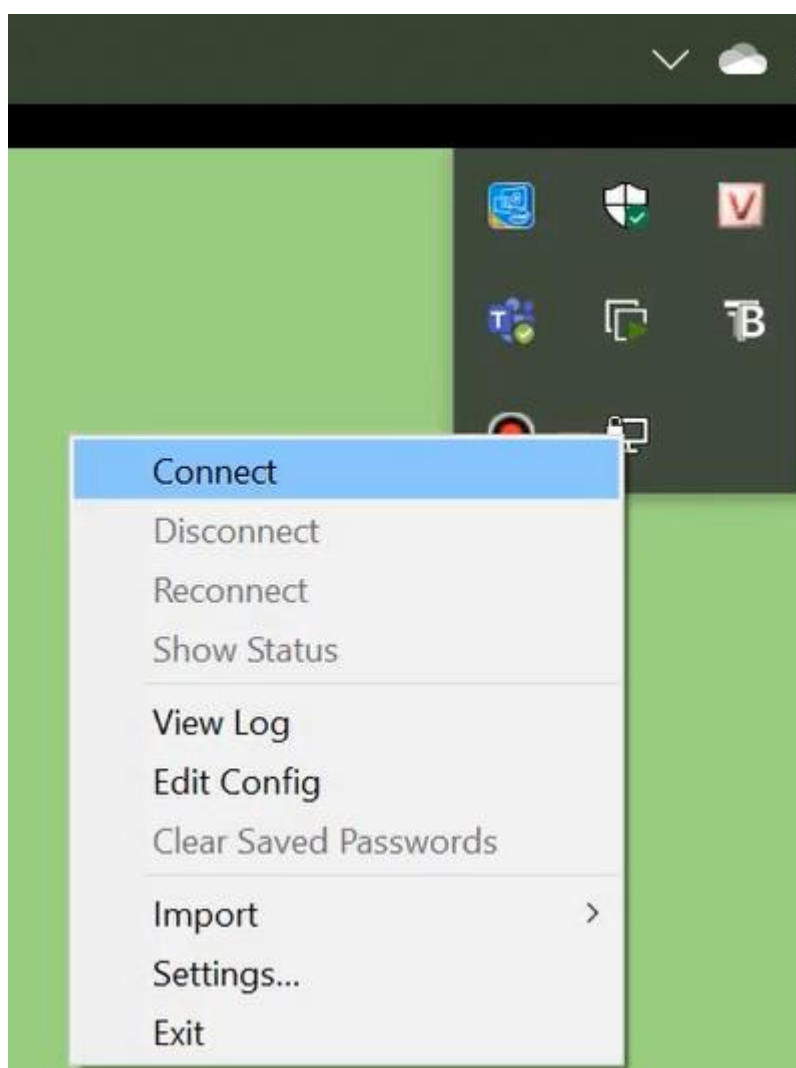
C:\Program Files\OpenVPN\config



Hình 25. Sao chép file *client1.ovpn* vào folder *config*

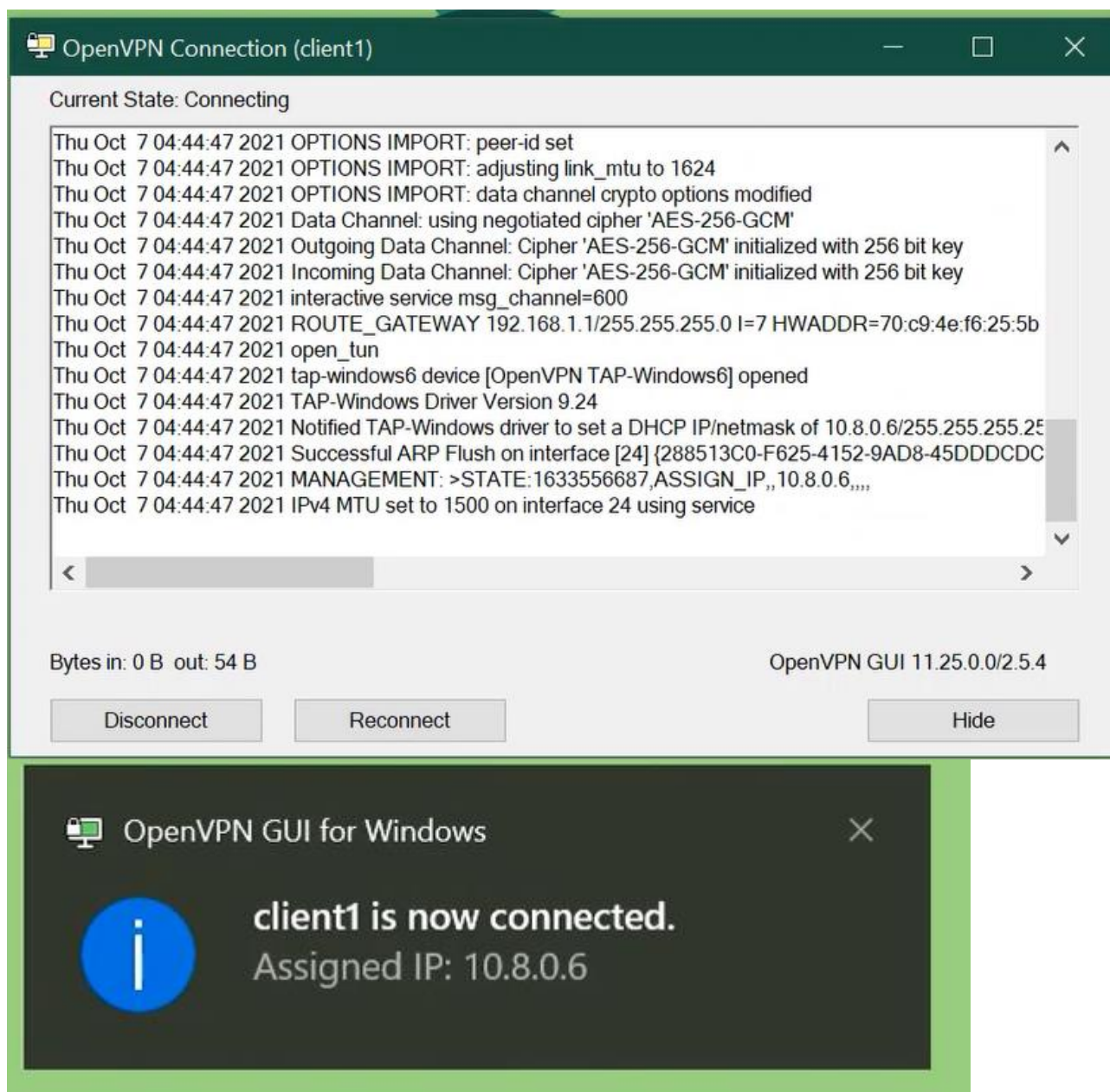
Khi chạy OpenVPN, nó sẽ tự động định vị profile và cung cấp profile đó.

Sau khi OpenVPN được khởi động, bắt đầu kết nối bằng cách vào applet khay hệ thống và nhấp chuột phải vào biểu tượng applet OpenVPN. Thao tác này sẽ mở menu ngữ cảnh. Chọn *client1* ở đầu menu (đó là profile *client1.ovpn*) và chọn *Connect*.



Hình 26. Connect OpenVPN

Một cửa sổ trạng thái sẽ mở ra hiển thị kết quả log trong khi kết nối được cài đặt và một thông báo sẽ hiển thị khi client được kết nối.

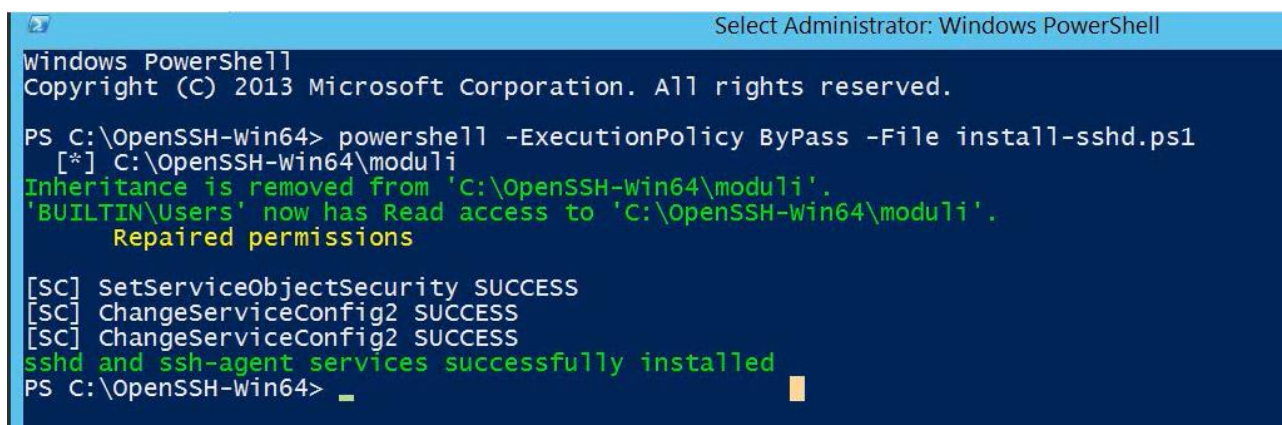


Hình 27. Kết nối OpenVPN thành công

2.2.2. Cấu hình SSH

• Bước 1: Tiến hành cài đặt OpenSSH trên Windown Server

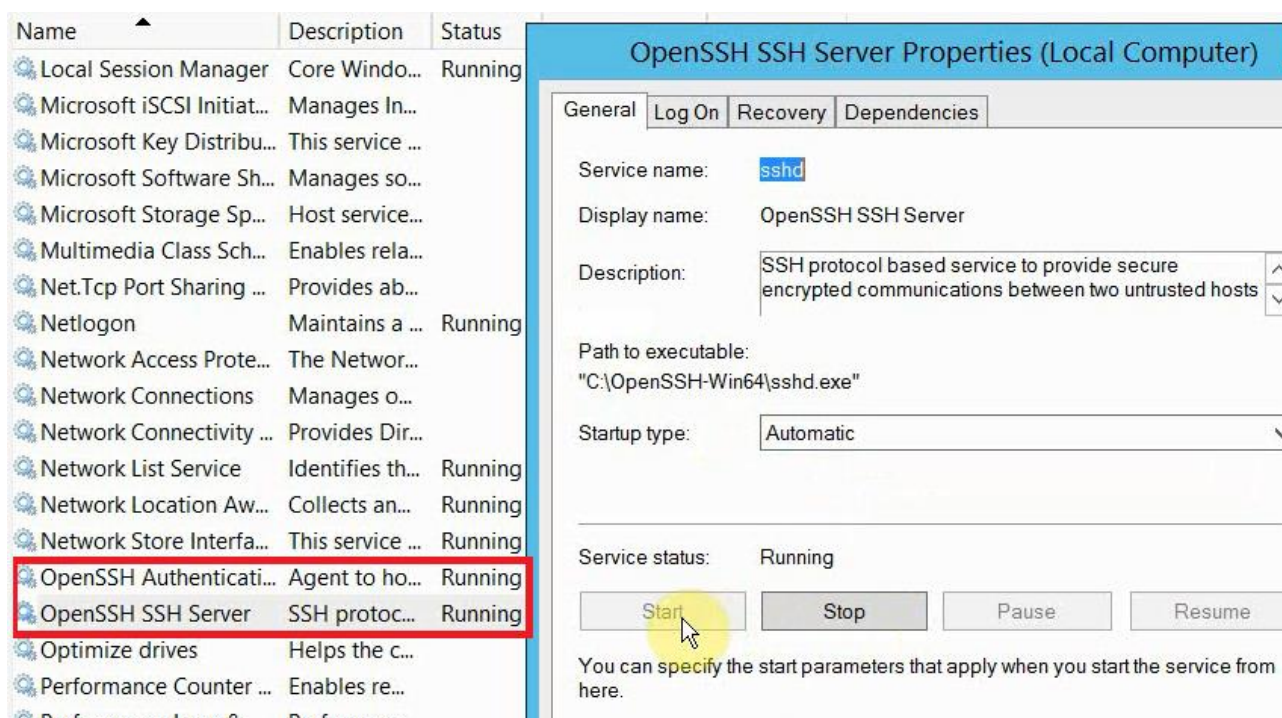
Sau khi tải file cài đặt OpenSSH trên Windown Server về, em tiến hành giải nén và cài đặt bằng PowerShell



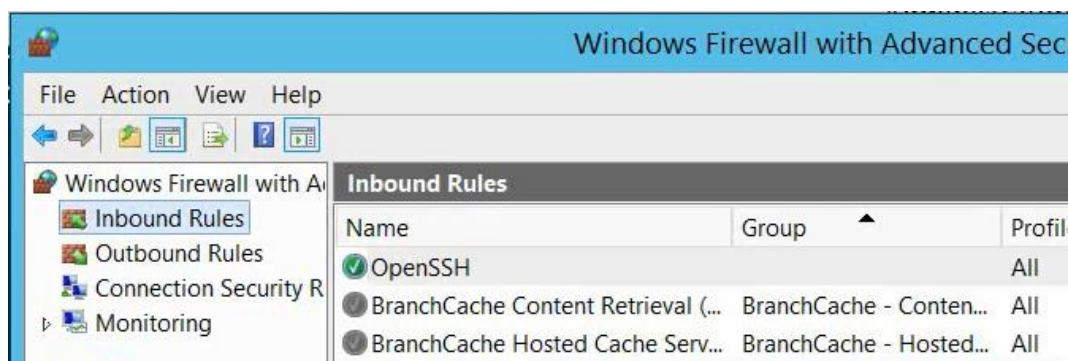
Hình 28. Cài đặt OpenSSH bằng PowerShell trên Window Server

• Bước 2: Chạy dịch vụ và cấu hình Firewall cho OpenSSH

Chạy 2 dịch vụ của OpenSSH lên bằng lệnh `service.msc` trong PowerShell



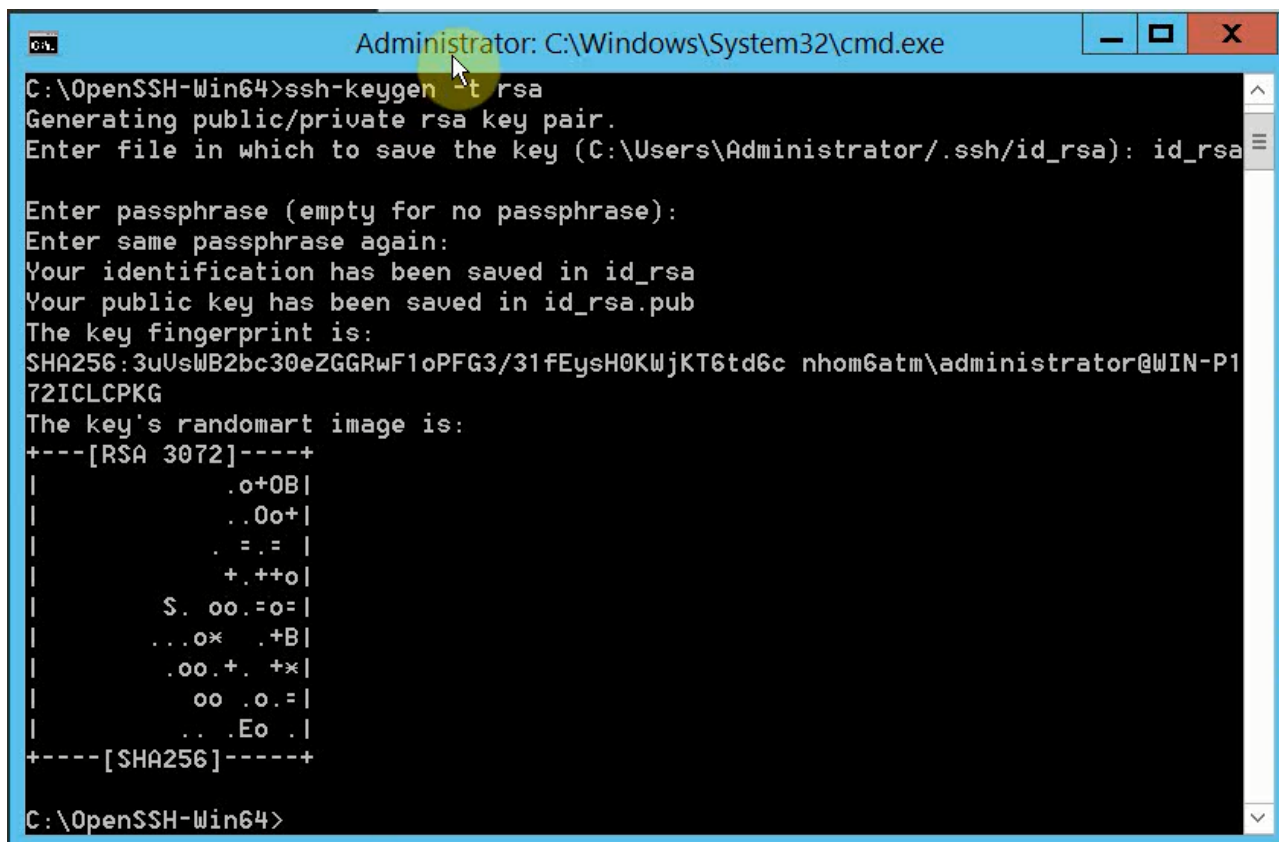
Hình 29. Chạy dịch vụ OpenSSH



Hình 30. Thêm Port 21 vào Inbound Rules trên Firewall

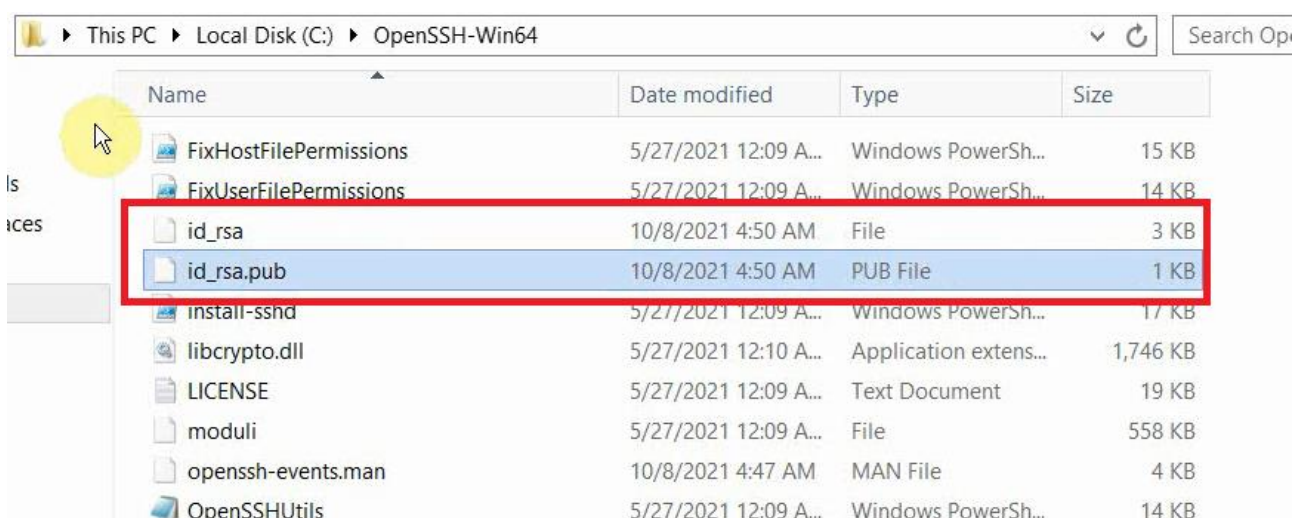
- **Bước 3: Tạo SSH key**

Mở cmd → chạy lệnh `ssh-keygen -t rsa` để tạo key SSH



Hình 31. Tạo key SSH

Sau khi tạo key SSH thu được file `id_rsa` (private key) và `id_rsa.pub` (public key)

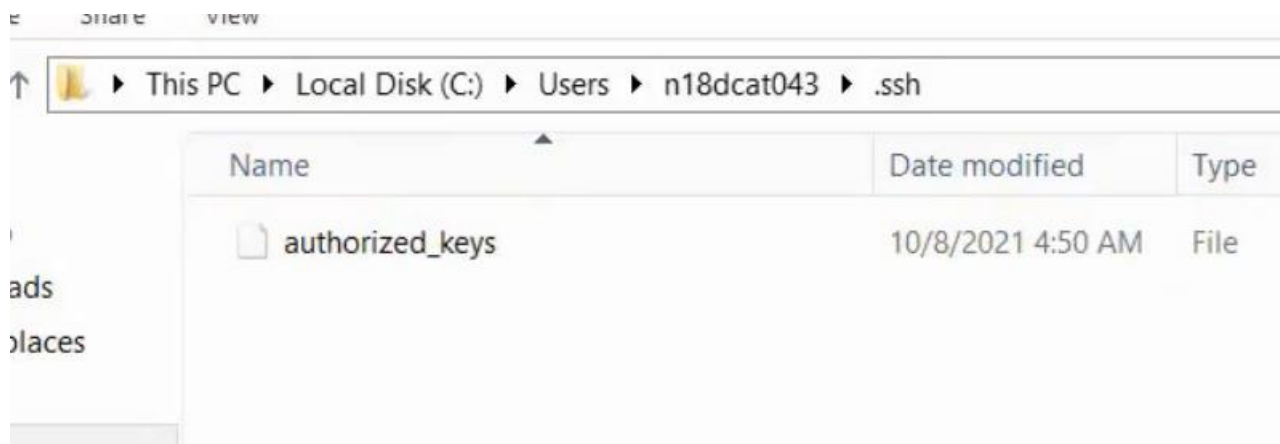


Hình 32. Cập key SSH vừa được tạo

- **Bước 4: Copy file Public Key vào Folder .ssh của account phân quyền cho phép truy cập**

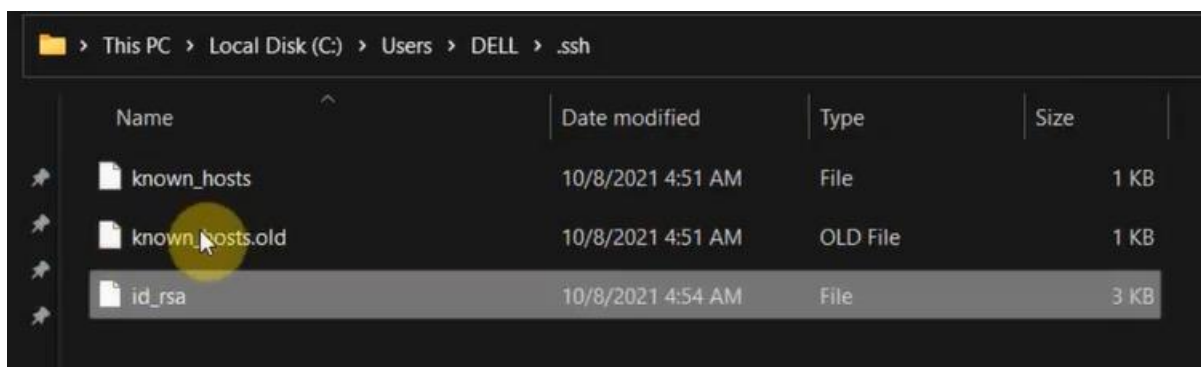
Copy file Public Key vào Folder .ssh của các user n18dcat043 và n18dcat091

Public Key đã được đổi tên thành *authorized_keys* (tên cũ là *id_rsa.pub*)



Hình 33. Copy Public Key vào folder .ssh của user n18dcat043

• **Bước 5: Lưu Private Key vào máy trạm cần thực hiện kết nối SSH**



Hình 34. Lưu Private Key vào máy trạm cần thực hiện kết nối SSH

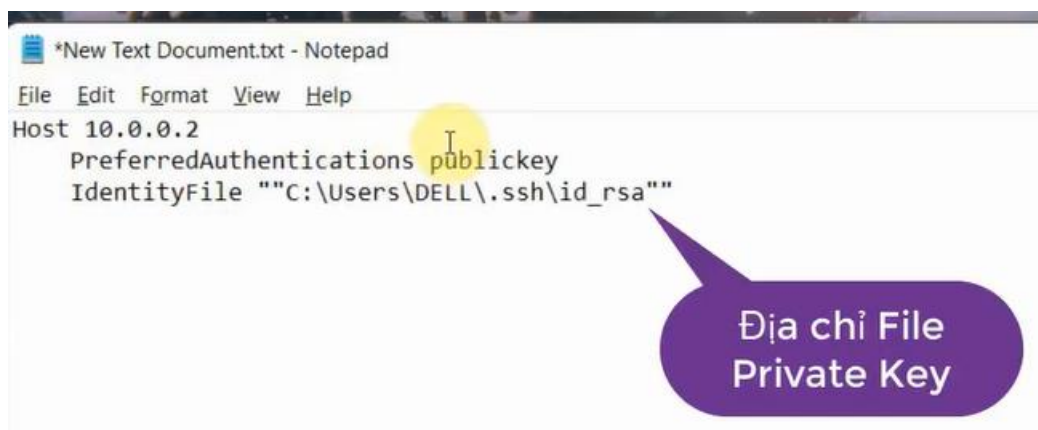
• **Bước 6: Khởi tạo file configs ssh**

Em tiến hành khởi tạo và chỉnh sửa file configs ssh như sau:

Host: địa chỉ truy cập host ssh

PreferredAuthentications: chế độ xác thực (ở đây ta sử dụng public key)

IdentityFile: địa chỉ lưu giữ private key



Hình 35. Chỉnh sửa file config ssh

• **Bước 7: Chỉnh sửa file cấu hình ssh trên máy Window Server**

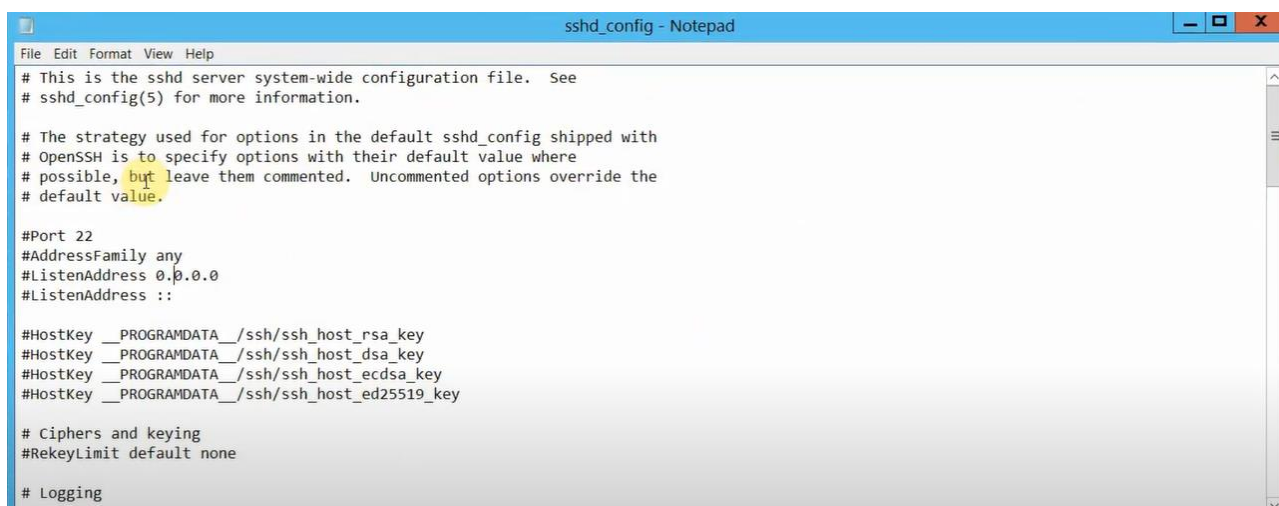
Em tiến hành chỉnh sửa file *sshd_config* trên máy Window Server với các thông số như sau:

Port 22 (Cổng kết nối)

PubkeyAuthentication yes (Bật chế độ xác thực SSH key)

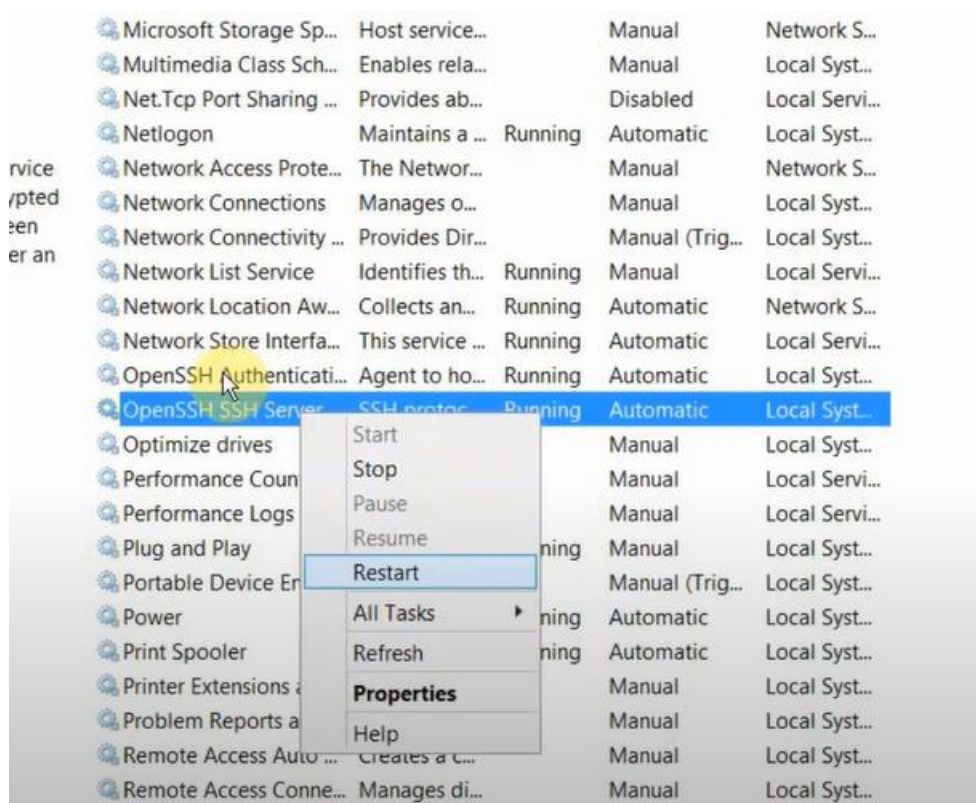
AuthorizedKeysFile .ssh/authorized_keys (Chỉ ra đường dẫn lưu Public key ví dụ lưu tại *.ssh/authorized_keys*)

PasswordAuthentication no (Tắt xác thực Password)



Hình 36. Chỉnh sửa file cấu hình sshd_config

• **Bước 8: Restart service SSH sau khi chỉnh sửa file cấu hình**



Hình 37. Restart server SSH

3. KẾT QUẢ NHẬN XÉT

3.1. Thực hiện kiểm tra kết nối OpenVPN

Từ một máy bên ngoài mạng OpenVPN, sau khi kết nối OpenVPN em thu được kết quả kết nối thành công

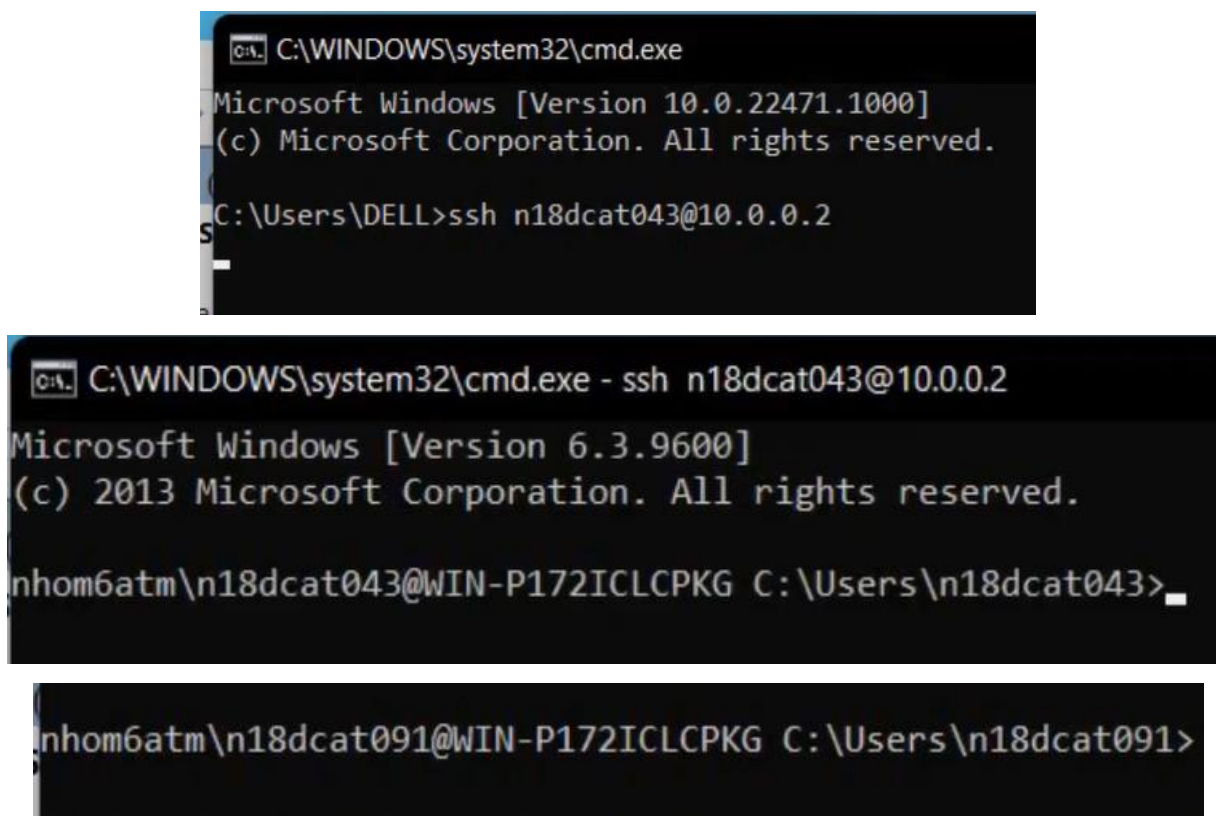
```
C:\Users\vtht1>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:
Reply from 10.0.0.2: bytes=32 time=10ms TTL=128
Reply from 10.0.0.2: bytes=32 time<1ms TTL=128
Reply from 10.0.0.2: bytes=32 time<1ms TTL=128
```

Hình 38. Ping thành công đến địa chỉ IP của hệ thống mạng bên trong OpenVPN

3.2. Thực hiện kiểm tra kết nối SSH

Từ một máy bên ngoài đã kết nối OpenVPN, em tiến hành kết nối SSH đối với user *n18dcat043* và *n18dcat091* → thu được kết quả truy cập thành công



Hình 39. Kiểm tra kết nối ssh đối với user *n18dcat043* và *n18dcat091*

Tương tự, kiểm tra kết nối SSH đối với user *n18dcat095* → thu được kết quả truy cập thất bại vì user này không được phân quyền truy cập

The screenshot shows a Windows command prompt window with the command 'ssh n18dcat095@10.0.0.2' entered. The output shows the connection attempt to 'n18dcat095@10.0.0.2' resulting in 'Permission denied (publickey,keyboard-interactive)'.

Hình 40. User *n18dcat095* truy cập SSH thất bại

3.3. Tác dụng EasyRSA trong OpenVPN

EasyRSA là một tiện ích CLI có chức năng chính là thiết lập Certificate Authority (CA) nội bộ dùng cho VPN server, VPN client và quản lý PKI CA.

OpenVPN là một VPN TLS / SSL, nghĩa là nó sử dụng các certificate để mã hóa lưu lượng giữa server và client. Do đó ta cần EasyRSA cấp các certificate này.

Dù là 2 project riêng biệt, nhưng EasyRSA được phát triển song song với OpenVPN. Do đó EasyRSA không thể thiếu hoặc bị thay thế trong tiến trình cài đặt OpenVPN.

4. PHÂN CÔNG TỰ ĐÁNH GIÁ

4.1. Phân công:

1. Võ Thị Hoa Tranh – N18DCAT091: Thực hiện cấu hình hệ thống OpenVPN trên Ubuntu Server và kiểm tra kết nối VPN
2. Lệnh Hà Bảo Long – N18DCAT043:
 - Thực hiện cấu hình OpenSSH trên Windown Server và phía client
 - Thực hiện phân quyền truy cập SSH bằng Public Key và Private Key
3. Lệnh Hà Bảo Trọng – N18DCAT095:
 - Chuẩn bị cơ sở lý thuyết, tìm kiếm tài liệu
 - Xây dựng báo cáo và làm video

4.2. Tự đánh giá:

Sau khi thực hiện đồ án, chúng em đã hiểu được cơ bản OpenVPN là gì, cách cài đặt, cấu hình và hoạt động ra sao. Trong quá trình học hỏi và nghiên cứu vẫn còn nhiều thiếu sót, cũng như việc học online hạn chế điều kiện để chúng em được tìm hiểu thực tế và sâu rộng. Chúng em cảm ơn cô vì đã hướng dẫn, giải đáp tận tình các câu hỏi để đồ án được thực hiện tốt hơn.