Báo cáo thực thực tập công ty an ninh mạng viettel: Firewall và VPN

Bùi Hoàng Dũng

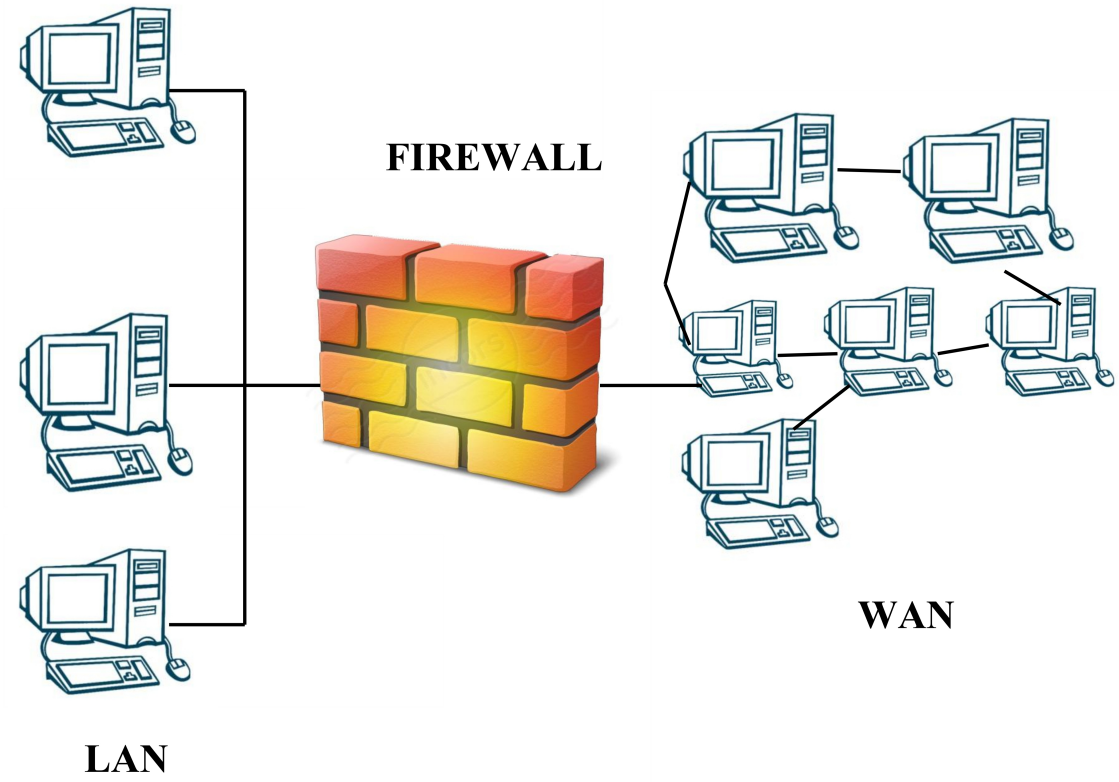
February 2024

Mục lục

1 Firewall

1.1 Firewall là gì

Firewall là một thiết bị bảo mật có khả năng lọc những traffic đến và traffic đi trong một private network. Lấy một ví dụ thực tế khi mình muốn thăm một người bạn của mình ở một khu trung cư, đầu tiên mình phải được bảo vệ ở đó cấp quyền cho đi lại (bằng cách đeo thẻ hay gì đó), người bảo vệ sẽ check với người bạn của mình rằng mình có phải bạn của bạn mình hay không. Nếu được chấp nhận mình sẽ được cấp quyền vào còn những người khác không được chấp nhận sẽ không có quyền vào. Việc được cấp quyền hay không phụ thuộc vào người bạn của mình. Trong trường hợp kể trên người bảo vệ sẽ có nhiệm vụ giống như là một firewall. Trong một hệ thống mạng thức tế, firewall đóng vai trò như là một người 'giữ cổng' ở computer's entry khi chỉ nhận những traffic mà được cấu hình để chấp nhận. Firewall sẽ thực hiện quá trình lọc network traffic trong mạng và phân tích traffic nào được cho phép forwarding còn traffic nào bị chặn dựa trên hoạt động tấn công mạng hay xảy ra nhằm ngăn chặn hệ thống bị tấn công mạng.



Hình 1: Firewall

Một firewall sẽ có 3 mục đích:

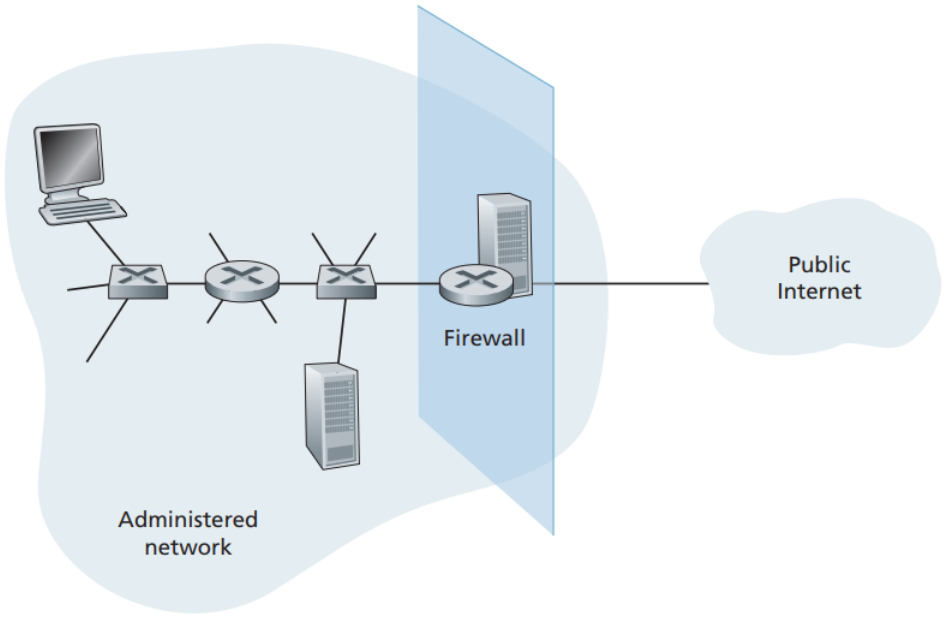
1. Tất cả traffic từ bên ngoài đến bên trong đều phải đi qua nó. Firewall sẽ nằm giữa phần mạng có thể được quản lý và internet. Một hệ thống lớn có thể sử dụng nhiều firewall cùng lúc.
2. Chỉ cho những traffic đã được xác thực (được định nghĩa ở trong security policy) đi qua.
3. Firewall là một thiết bị giúp cho hệ thống miễn nhiễm với sự xâm nhập.

1.2 Phân loại

Firewall được phân ra làm 3 loại:

- Traditional packet filters
- state-ful filter
- application gateway

1.2.1 Traditional packet filters



Hình 2: Topo

Một tổ chức sẽ luôn có một gateway router để kết nối với mạng nội bộ với Internet. Tất cả các traffic đi và đến trong mạng nội bộ này đều được thông qua chiếc router này, không những vậy router này có có một chức năng gọi là **packet filtering**. Packet filter sẽ kiểm tra dữ liệu đến hoặc đi và xác định xem lưu lượng này sẽ được cho phép đi đến đâu hoặc có thể bị dropped dựa trên rules do người quản trị đặt ra. Filter decision sẽ thường được dựa trên:

- Địa chỉ IP đích và nguồn
- Giao thức sử dụng (TCP, UDP, ICMP, OSPF,....)
- Địa chỉ port đích và nguồn của TCP hoặc UDP
- Loại tin nhắn ICMP
- Những rules khác nhau cho traffic đến và đi
- Những rules khác nhau được đặt cho các interface khác nhau của router

Người quản trị mạng sẽ cấu hình cho firewall dựa trên policy của tổ chức. Policy này có thể là băng thông giới hạn được sử dụng cho mỗi account của tổ chức để gia tăng vấn đề về bảo mật.

Policy	Firewall Setting
No outside Web access.	Drop all outgoing packets to any IP address, port 80.
No incoming TCP connections, except those for organization's public Web server only.	Drop all incoming TCP SYN packets to any IP except 130.207.244.203, port 80.
Prevent Web-radios from eating up the available bandwidth.	Drop all incoming UDP packets—except DNS packets.
Prevent your network from being used for a smurf DoS attack.	Drop all ICMP ping packets going to a "broadcast" address (eg 130.207.255.255).
Prevent your network from being tracerouted.	Drop all outgoing ICMP TTL expired traffic.

Hình 3: Filter policy

Hình trên miêu tả một số policy và poclicy đó được cấu hình như thế nào ở trên firewall. Ví dụ như là nếu tổ chức không muốn có kết nối TCP đến ngoại trừ đến với public Web server của tổ chức, thì nó có thể chặn tất cả TCP SYN segment đến ngoại trừ TCY SYN segment port đích là 80 và địa chỉ IP đích là địa chỉ IP của Web server. Nếu mà tổ chức không muốn user sử dụng băng thông cho Internet radio application thì nó có thể chặng tốt cả các gói tin UDP đến ngoại trừ gói tin DNS.
Một filter policy có thể được dựa trên sự kết hợp của địa chỉ và port. Ví dụ một filtering router có thể forward tất cả gói tin Telnet (port 23) ngoại trừ gói tin có địa chỉ đến và địa chỉ nguồn từ một danh sách các địa chỉ IP đặc biệt. Policy này cho phép kết nối telnet đến từ host có địa chỉ nằm trong danh sách được cho phép. Tuy nhiên việc này lại không thể nào chống lại được tấn công với những gói tin có địa chỉ giả mạo. Filtering có thể dựa trên việc TCK ACK bit được set hay là không. Điều này khá hiệu quả khi ta muốn cho client bên trong có thể kết nối với server bên ngoài nhưng lại ngăn chặn client bên ngoài muốn kết nối với client bên trong. Điều này có thể được lý giải như sau: trong segment đầu tiên của mỗi TCP connection đều có ACK bit được set là 0, trong khi tất cả các segment khác trong kết nối đều có ACK bit được set là 1, Do đó nếu tổ chức muốn ngăn chặn client bên ngoài khỏi việc thiết lập kết nối tới server bên trong thì nó chỉ việc lọc các TCP segment đến có ACK bit set là 0. Policy này kill tất cả các kết nối TCP bắt nguồn từ bên ngoài và cho phép thiết lập kết nối TCP bắt nguồn từ bên trong. Firewall rules được trển khai trên router với ACL (access control list) với mỗi interface của router sẽ có một ACL riêng cho mình.

action	source address	dest address	protocol	source port	dest port	flag bit
allow	222.22/16	outside of 222.22/16	TCP	> 1023	80	any
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53	—
allow	outside of 222.22/16	222.22/16	UDP	53	> 1023	—
deny	all	all	all	all	all	all

Hình 4: Access Control List

Hình trên mô tả một ACL trên một router interface với một mạng có subnet là **222.22/16**. ACL này được áp dụng cho interface của router có kết nối với mạng bên ngoài. Các rules được áp dụng cho mỗi gói tin theo thứ tự ưu tiên từ trên xuống dưới. 2 rules đầu tiên cho phép user bên trong được surf web. Lưu ý rằng khi một source bên ngoài muốn thiết lập kết nối TCP với một host bên trong thì kết nối này sẽ bị chặn mặt dù source hoặc destination port của nó là 80. 2 rules tiếp theo cho phép gói tin DNS đến và đi vào mạng.

1.2.2 Stateful Packet Filter

Trong traditional packet filet thì quyết định lọc được thực hiện dựa trên từng gói trong mạng. Stateful filter có thể dựa vào thông tin về kết nối TCP và dùng thông tin đó để thực hiện quyết định lọc. Ví dụ như trong bảng ở phân trước, dẫu biết là rules cho phép những packet đến tư bên ngoài với ACK bit bằng 1 và source port là 80 được đi qua filter. Những packet này có thể được sử dụng bới attacker để làm hư hại hệ thống bên trong. Cách đầu tiên mà ta có thể nghĩ ra là chặn tất các TCP ACK traffic nhưng điều này sẽ ngăn chặn các TCP ACK từ mạng bên trong. Stateful filter sẽ giải quyết vấn đề này bằng cách theo dõi tất cả các kết nối TCP đến trên connection table. Điều này khả thi vì firewall có thể quan sát sự bắt đầu của một kết nối TCP bằng cách nhìn three-way handshake (SYN, SYNACK và ACK) và nó có thể quan sát kết thúc của một kết nối khi nó nhìn thấy FIN packet. Firewall có thể giả định rằng kết nối này đã hết khi nó không nhìn thấy bất cứ hoạt động nào của kết nối này nữa (60s chẳng hạn).

source address	dest address	source port	dest port
222.22.1.7	37.96.87.123	12699	80
222.22.93.2	199.1.205.23	37654	80
222.22.65.143	203.77.240.43	48712	80

Hình 5: Connection table

Trong bảng này có 3 kết nối TCP đang được thực hiện, và tất cả chúng đều được tạo bởi client bên trong. Stateful filter sẽ thông một cột mới **check connection** vào access control list của nó như sau:

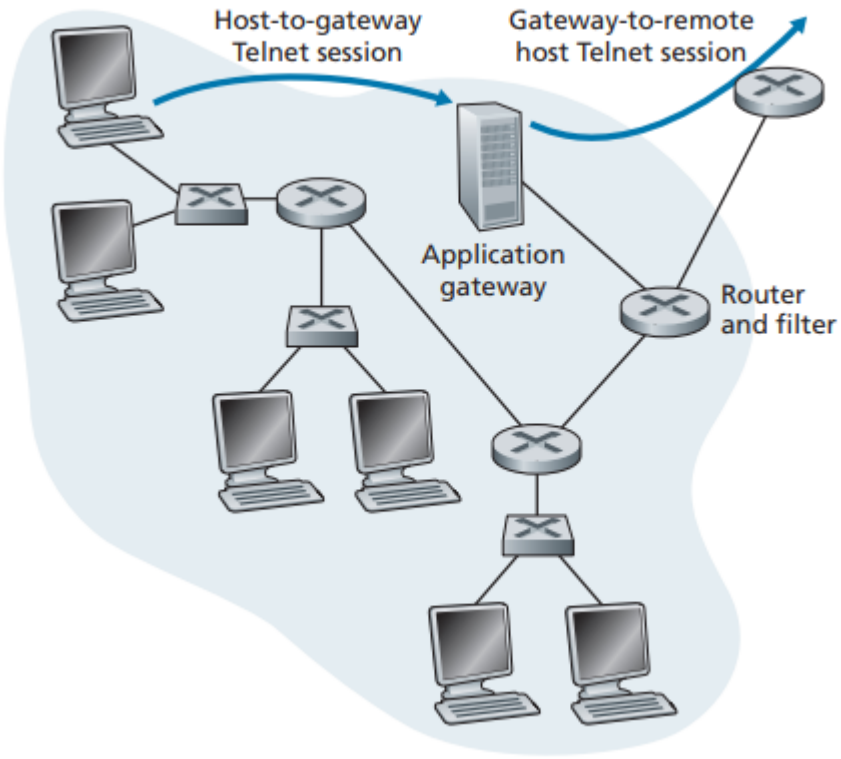
action	source address	dest address	protocol	source port	dest port	flag bit	check conxion
allow	222.22/16	outside of 222.22/16	TCP	> 1023	80	any	
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK	X
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53	—	
allow	outside of 222.22/16	222.22/16	UDP	53	> 1023	—	X
deny	all	all	all	all	all	all	

Hình 6: ACL cho stateful filter

Giả sử khi một gói tin TCP với source port là 12543 và địa chỉ IP đích là 150.23.23.155. Khi gói tin này đến firewall thì firewall sẽ kiểm tra trong ACL và ở bước này firewall cũng sẽ kiểm tra connection table và nhận thấy rằng gói tin này không nằm trong danh sách cách TCP connection đang được thực hiện và block luôn gói tin này. Ví dụ thứ hai, cho rằng người dùng bên trong muốn surf Website ở bên ngoài mạng, bởi vì người dùng này phải thiết lập kết nối TCP đầu tiên nên là TCP connection của user này sẽ được ghi lại trong connection table. Khi Web server response lại với gói tin có bit ACK được set, firewall sẽ kiểm tra bảng và nhìn thấy rằng kết nối này đang trong quá trình thực hiện và để cho gói tin này đi qua.

1.3 Application Gateway

Trong 2 loại firewall trước, ta đã thấy được firewall thực hiện lọc gói tin như thế nào dựa trên các trường dữ liệu của TCP/UDP header và IP header. Tuy nhiên nếu mà tổ chức muốn cung cấp dịch vụ chẳng hạn như Telnet chỉ cho một số người dùng bên trong. Để có một lớp bảo mật tốt hơn, firewall phải kết hợp packet filter với application gateway. Application không nhìn vào IP header hay TCP/UDP header mà lên quyết định dựa trên dữ liệu của ứng dụng. Một application gateway được coi như là một application server đặc biệt nơi mà tất cả dữ liệu của ứng dụng (dữ liệu đến và đi) phải đi qua. Nhiều application gateway có thể chạy trên một host, nhưng mỗi gateway được chia ra là một server riêng với các hoạt động riêng. Để thấy được rõ hơn, ta sẽ thực hiện thiết kế một firewall có thể cho phép một danh sách các user bên trong có thể telnet ra bên ngoài và ngăn chặn các telnet từ bên ngoài vào bên trong. Policy này có thể được hoành thành bằng cách kết hợp packet filter (trong router) và một telnet application gateway.



Hình 7: Firewall kết hợp của packet filter và application gateway

Router filter sẽ được cấu hình để chặn tất cả các Telnet connection ngoại trừ kết nối bắt nguồn từ địa chỉ IP của application gateway. Filter này sẽ được cấu hình chặn tất cả các Telnet connection từ bên ngoài đi qua application gateway. Nếu một user bên trong muốn Telnet ra bên ngoài thì user đó phải set up một Telnet session với application gateway. Một application đang chạy trong gateway sẽ nghe những Telnet session đến và hiện cho user một prompt để nhập username và password. Khi user nhập xong, application gateway sẽ kiểm tra xem user này có quyền Telnet ra bên ngoài hay không. Nếu không Telnet connection từ user sẽ bị hủy bởi gateway. Nếu có thì gateway sẽ hiện cho user thấy được hostname của host mà user muốn Telnet tới, thực hiện việc chuẩn bị một phiên Telnet giữa gateway và host bên ngoài và chuyển tất cả dữ liệu đến từ user đến host bên ngoài. Ngoài ra Telnet application gateway không chỉ thực hiện việc xác thực người dùng mà còn hoạt động như một Telnet server và một Telnet server để chuyển tiếp dữ liệu từ user đến host bên ngoài. Trong thực tế thường sẽ có nhiều application gateway ví dụ như là gateway cho Telnet, HTTP, FTP và e-mail. Tuy nhiên application gateway vẫn tồn tại một số nhược điểm như là cần cho mỗi application, performance của mạng sẽ phải trả giá vì có thể sẽ có nhiều user và application dùng chung một gateway.

1.4 Triển khai firewall

1.4.1 Phân Zone

Dựa vào đặc tính của từng host mà khi triển khai firewall ta sẽ phân các host đó ra thành từng zone một

- Internal Zone (Vùng nội bộ): bao gồm các tài nguyên nội bộ và máy tính của người dùng. Trong

internal zone ta có thể chia thành User Zone và Server Zone.

- DMZ: là vùng chứa các dịch vụ công cộng hoặc các ứng dụng có thể tiếp xúc với mạng bên ngoài. Các web server. email hoặc các dịch vụ có liên quan.
- External Zone là vùng mà đại diện cho môi trường mạng bên ngoài tổ chức. Các kết nối Internet, VPN từ bên ngoài đều được xử lý trong vùng này

Ngoài việc chia các host ra thành từng zone, khi triển khai firewall còn thực hiện việc đưa ra những policy để cấu hình firewall:

- Quy tắc từ nội bộ ra ngoại bộ: Cho phép các máy tính ở internal zone được truy cập internet. Kiểm soát các loại lưu lượng tư bên ngoài (HTTP, HTTPs, DNS) để đảm bảo an toàn.
- Quy tắc từ ngoại bộ vào nội bộ: Kiểm soát các kết nối đến các dịch vụ nội bộ, như email server, web server, hạn chế truy cập dựa trên địa chỉ IP và cổng.
- Quy tắc từ nội bộ ra DMZ: Cho phép các ứng dụng nội bộ giao tiếp với các dịch vụ ở DMZ, giữ cho lưu lượng này được kiểm soát chặt chẽ.
- Quy tắc Từ DMZ vào Nội bộ: Kiểm soát việc các dịch vụ ở DMZ có thể gửi thông tin vào mạng nội bộ. Hạn chế rủi ro từ các dịch vụ nguy cơ.
- Quy tắc Logging và Monitoring: Bật chức năng log cho mọi quy tắc để theo dõi và phân tích sự kiện.
- Quy tắc Emergency: Thiết lập quy tắc để ngăn chặn ngay lập tức trong trường hợp xâm nhập hoặc mối đe dọa nguy cơ cao.

1.5 Thực hành trên Iptable

Iptables là một software application firewall trên Linux. Network traffic được tạo từ các packets. Dữ liệu được chia nhỏ ra thành từng mảnh gửi qua mạng được sắp xếp lại ở bên thu. Iptables định nghĩa việc các packet được nhận như thế nào thông qua một danh sách các rules để quyết định xem hệ thống nên làm gì với các packet đó. Các định nghĩa trong Iptables:

- Tables: Tables là các file có cùng tham gia vào hoạt động giống nhau. Một tables sẽ bao gồm nhiều **chain**.
- Chain: Một chain là một xâu ký tự biểu diễn **rules**. Khi một packet được nhận iptables tìm những tables và chạy nó thông qua một chuỗi các rules đến khi nào tìm thấy khớp thì thôi.
- Rules: Rules sẽ nói cho hệ thống phải làm gì với những packets. Rules có thể chặn một loại packet cũng có thể forward một loại packet. Đầu ra nơi mà packet được gửi đến sẽ được gọi là **target**.
- Target: Một target là một quyết định nên làm gì với một packet thường là drop, accecpt, reject(trong trường hợp packet lỗi).

Iptable trong linux sẽ mặc đinh là có 4 tables:

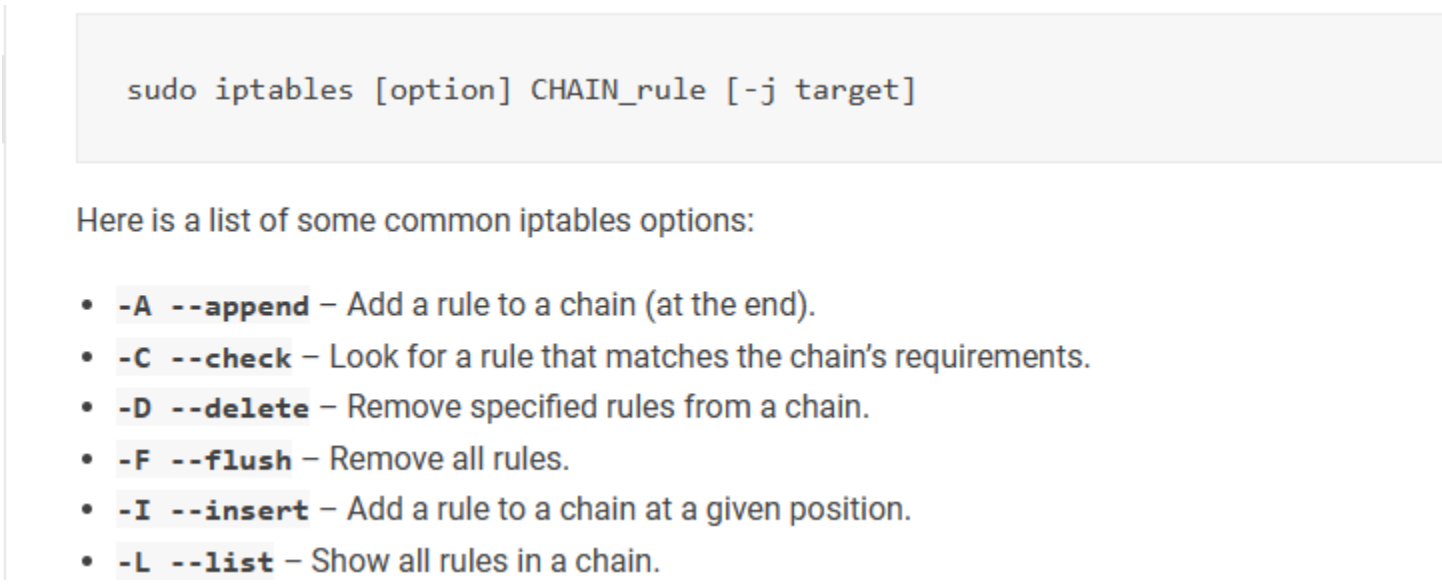
1. Filter: là bảng hay được sử dụng, quyết định xem packet nào được đi vào hoặc đi ra khỏi mạng. Filter có 3 chain đó là: **Input**(Rule trong chain này điều khiển các packet nhận), **Output**(Rule trong chain này điều khiển các packet ra), **Forward**(Rule trong chain này điều khiển các packet được điều hướng để đi qua server).

2. Network Address Translation (NAT): bảng này chứa các NAT rules cho việc routing các packet đến các mạng mà không thể kết nối trực tiếp. Khi mà địa chỉ nguồn và đích của gói tin bị thay đổi thì NAT table được sử dụng. Nó bao gồm các chain như sau: **Prerouting**(Chain này sẽ được gán cho packet ngay sau khi server nhận được nó), **Output**(Hoạt động như output chain trong filter table), **Postrouting**(rules trong chain này cho phép thực hiện thay đổi đến packet sau khi chúng rời Output chain).

3. Mangle table thực hiện việc điều chỉnh IP header của các packets. Bảng này có tất cả các chain đã được nói ở trên như: Prerouting, Postrouting, Output, Input, Forward.

4. Raw được sử dụng để loại bỏ theo dõi packet. Raw table có 2 chain được nhắc đến trước đó là: Prerouting, Output.

Các option trong câu lệnh iptables:



Hình 8: Các option trong iptables

1.5.1 Thực hiện chặn traffic đến từ một địa chỉ IP

Sử dụng câu lệnh: *iptables -A INPUT -s 192.168.88.134 -j DROP*.

```
root@dung:/home/buidung# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:https
DROP       all  --  192.168.88.134       anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Hình 9: Show iptables

```
root@buidung:/home/buidung# ping 192.168.88.130
PING 192.168.88.130 (192.168.88.130) 56(84) bytes of data.
^C
--- 192.168.88.130 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4102ms
```

Hình 10: Kết quả

1.5.2 Thực hiện chặn icmp traffic đến từ một địa chỉ và cho phép icmp traffic ra đến địa chỉ đó

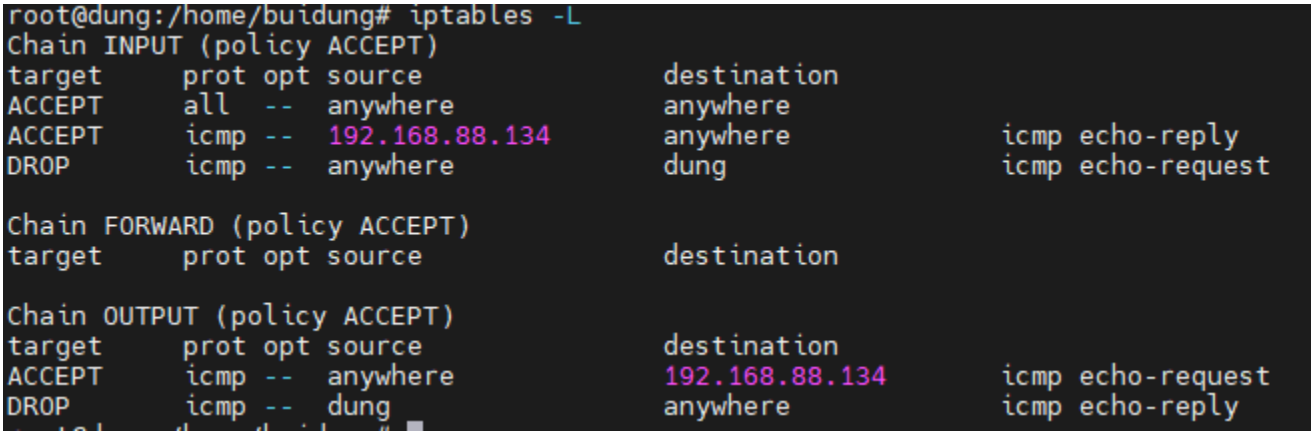
Thực hiện cho phép icmp traffic ra bên ngoài trên server:

- *iptables -A OUTPUT -p icmp –icmp-type echo-request -d 192.168.88.134 -j ACCEPT*
- *iptables -A INPUT -p icmp –icmp-type echo-reply -s 192.168.88.134 -j ACCEPT*

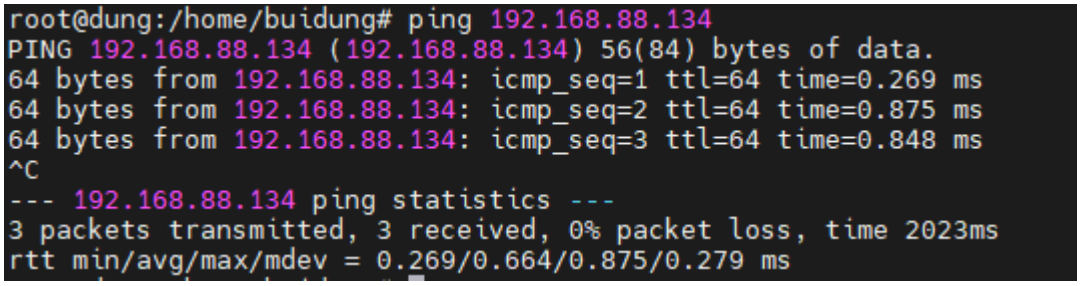
Thực hiện chặn icmp traffic đến trên server:

- *iptables -A INPUT -p icmp –icmp-type echo-request -d 192.168.88.130 -j DROP*
- *iptables -A OUTPUT -p icmp –icmp-type echo-reply -s 192.168.88.130 -j DROP*

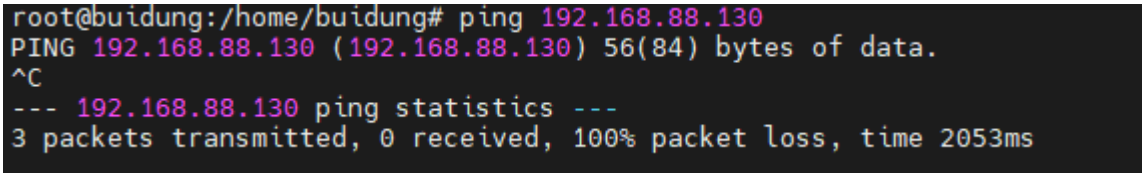
Sau khi thực hiện được bảng iptables như sau:



Hình 11: Show iptables



Hình 12: Ping from server



Hình 13: Ping to server

Lưu ý: Nếu muốn chặn địa chỉ IP trong một dải IP thì ta chỉ cần thay cờ *-s* và thay bằng cờ *-m* với module *iprange* và đưa ra dải địa chỉ IP với cờ *–src-range*.
Ví dụ: *sudo iptables -A INPUT -m iprange –src-range 192.168.1.100-192.168.1.200 -j DROP*.

1.5.3 Thực hiện config forwarding table

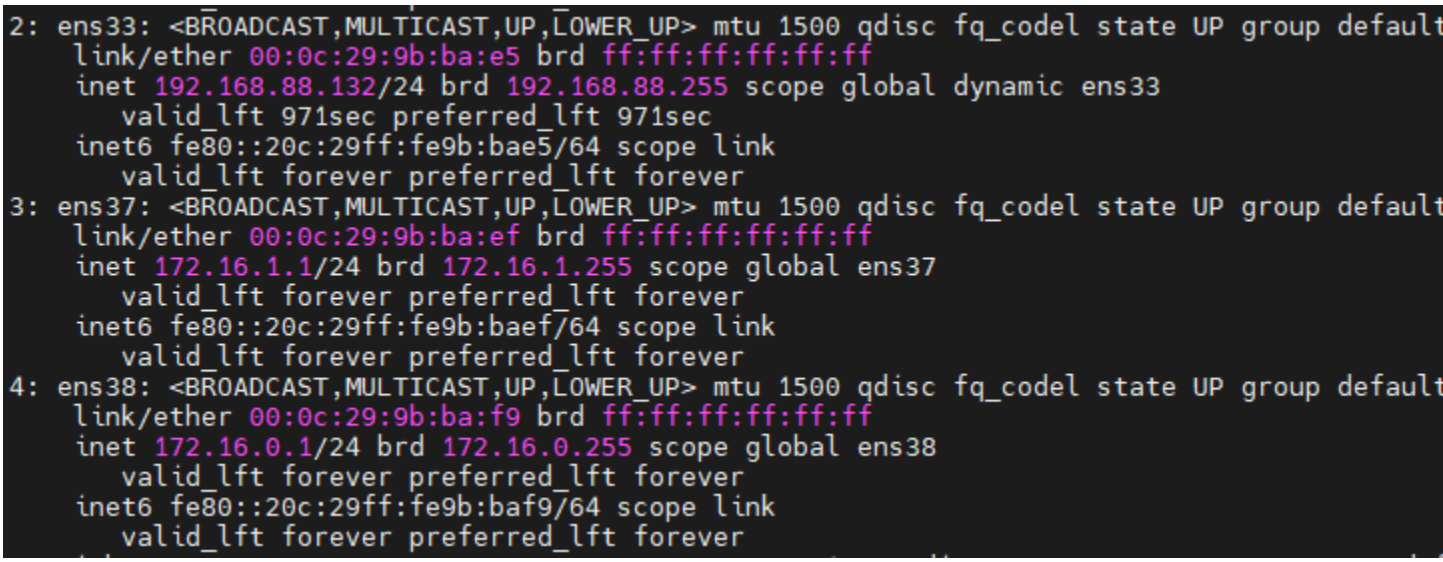
Ta có địa chỉ IP của host1 như sau:

Hình 14: Địa chỉ IP host1

Ta có địa chỉ IP của host2 như sau:

Hình 15: Địa chỉ IP host2

Ta có địa chỉ IP của router như sau:

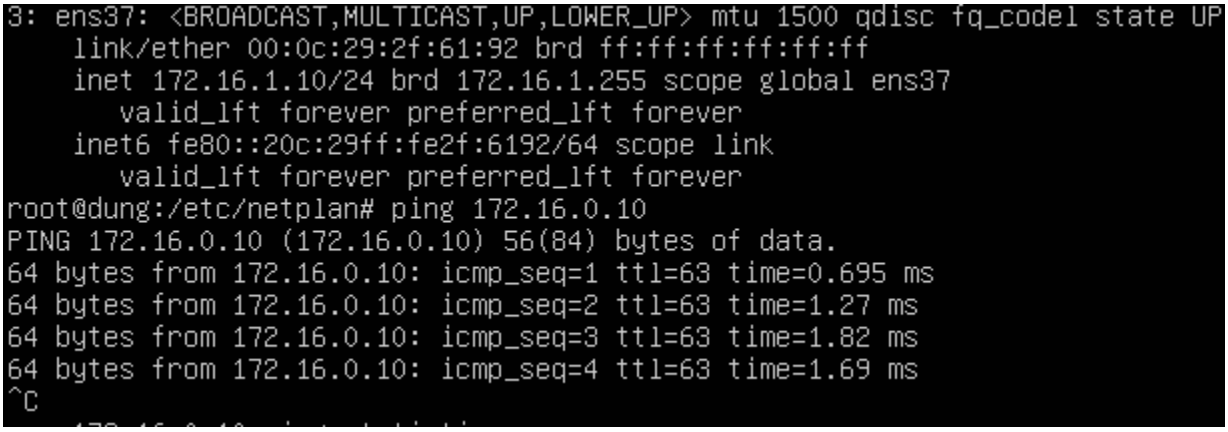


Hình 16: Địa chỉ IP của router

Set rules cho iptables:

- *iptables -A FORWARD -s 172.16.0.0/24 -d 172.16.1.0/24 -j ACCEPT*
- *iptables -A FORWARD -s 172.16.1.0/24 -d 172.16.0.0/24 -j ACCEPT*

Test ping giữa 2 host:

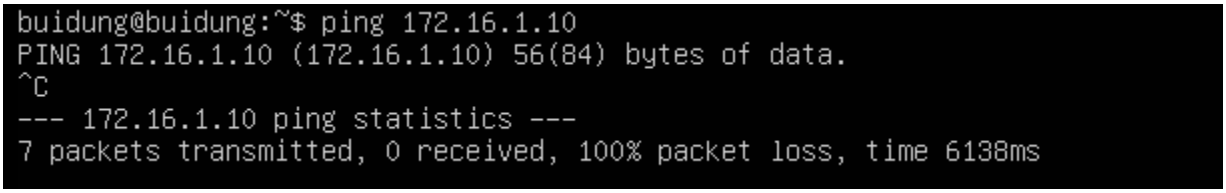


Hình 17: Test ping giữa 2 host

Thực hiện block traffic forward qua router:

- *iptables -A FORWARD -s 172.16.1.0/24 -d 172.16.0.0/24 -j DROP*
- *iptables -A FORWARD -s 172.16.0.0/24 -d 172.16.1.0/24 -j DROP*

Test ping giữa 2 host:



Hình 18: Test ping sau khi đã add rules DROP

2 VPN

2.1 VPN là gì ?

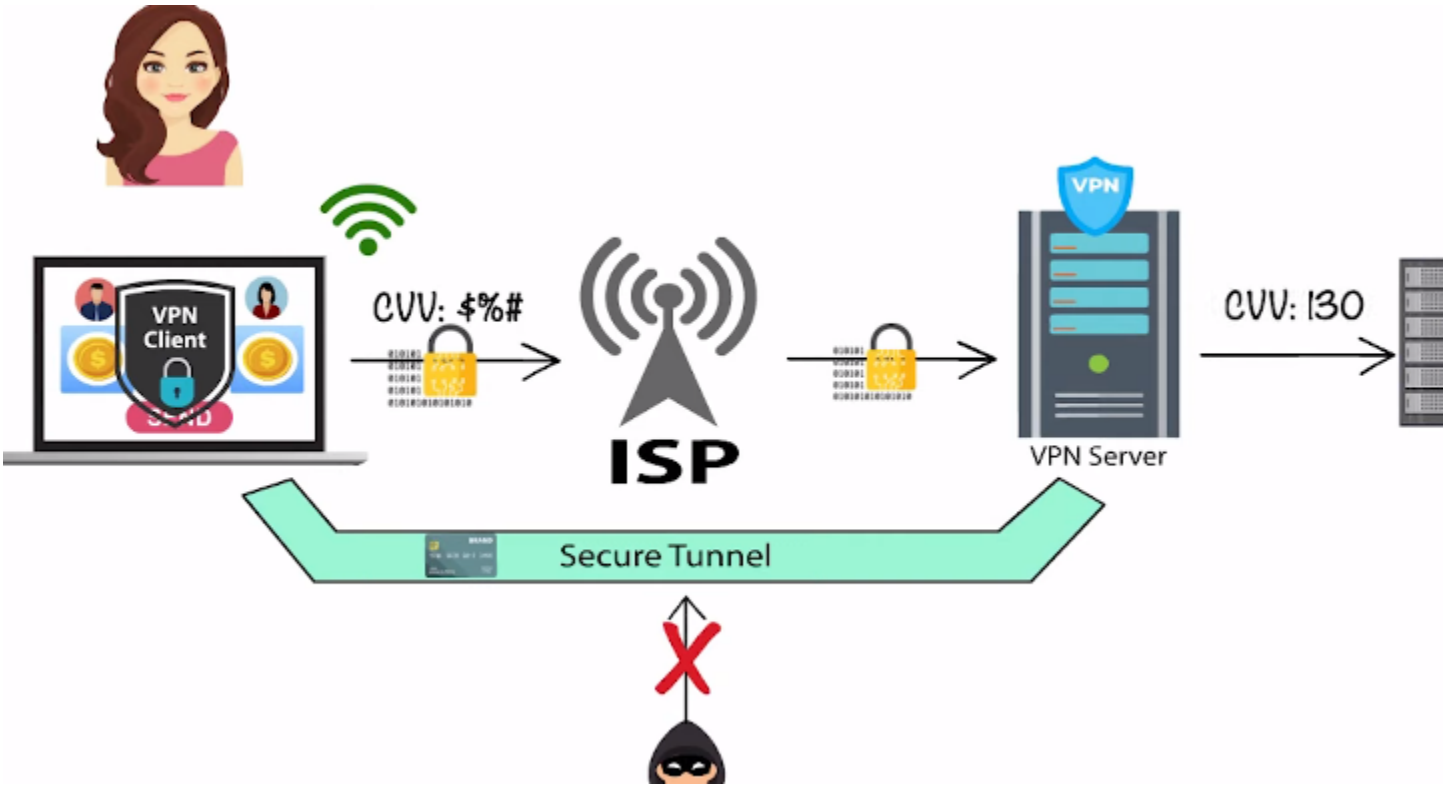
VPN hay còn gọi là Virtual Private Network là một kết nối được mã hóa qua internet từ một thiết bị đến một mạng máy tính. Kết nối mã hóa này giúp dảm bảo dữ liệu 'nhạy cảm' được truyền đi một cách an toàn. Nó ngăn chặn những người dùng không xác thực khỏi việc nghe trộm và cho phép người dùng thực hiện làm việc từ xa. VPN là một công cụ bảo mật được sử dụng giúp cho việc duyệt web một cách ẩn danh. Điều này được thực hiện bằng cách làm những điều như sau:

- **Mã hóa dữ liệu**: Những dữ liệu mã hóa này sẽ được nhìn thấy bởi bên thứ ba chẳng hạn như là trang web mà ta vừa mới vào, ISP mà ta thuê và những người tấn công mạng.
- **Ẩn địa chỉ IP**: Việc ẩn địa chỉ IP sẽ giúp cho địa chỉ thật của ta không bị tiết lộ và không ai có thể biết được ta đang ở đâu trong khi đang duyệt web mà sử dụng VPN.

VPN hoạt động được ở hầu hết các hệ điều hành như Window, Mac, Linux, Android và IOS.

2.2 VPN hoạt động như thế nào ?

Để sử dụng được VPN đầu tiên ta cần phải cài đặt **VPN client** trên máy tính hoặc điện thoại của mình. VPN client có khả năng thiết lập kết nối bảo mật để truyền dữ liệu thông qua internet. VPN client sẽ thực hiện việc kết nối đến với ISP (Internet Service Provider) và mã hóa những thông tin được gửi đi sử dụng VPN protocol. Sau đó VPN client thiết lập một VPN tunnel trong public network và kết nối đến VPN server. VPN tunnel sẽ bảo vệ thông tin được truyền đi bằng cách thay đổi địa chỉ IP và vị trí hiện tại của người dùng ở VPN. Cuối cùng VPN server sẽ kết nối với service server mà chúng ta cần. Ở bước này những dữ liệu sẽ được giải mã. Ở đây ta có thể thấy rằng VPN server sẽ được hoạt động như một proxy trên. Bằng cách giả mạo địa chỉ IP và vị trí hiện tại của người dùng VPN có thể được sử dụng để duyệt một trang web mà có thể bị chặn ở vị trí hiện tại của mình.



Hình 19: VPN work ?

VPN thực hiện mã hóa thông tin truyền qua đường hầm ảo bằng cách sử dụng các giao thức và thuật toán mã hóa.Sau đây sẽ là cách mà một VPN thực hiện mã hóa:

1. **Chọn phương thức mã hóa**: VPN sử dụng giao thức mã hóa để đảm bảo tính an toàn của dữ liệu. Các giao thức phổ biến bao gồm IPsec, SSL/TLS, PPTP và OpenVPN. Mỗi giao thức có cách thức hoạt động và cấp độ bảo mật khác nhau. IPsec thường được sử dụng cho kết nối site-to-site, trong khi SSL/TLS thường được sử dụng cho kết nối remote access và trình duyệt web.

2. **Quá trình Handshake**: Trước khi bắt đầu truyền dữ liệu, máy chủ VPN và thiết bị kết nối phải thực hiện quá trình handshake để thỏa thuận các tham số cho việc mã hóa. Trong quá trình này 2 bên sẽ chọn một phương thức mã hoá, chọn khóa mã hóa và thực hiện các bước cần thiết để thiết lập một liên kết an toàn.

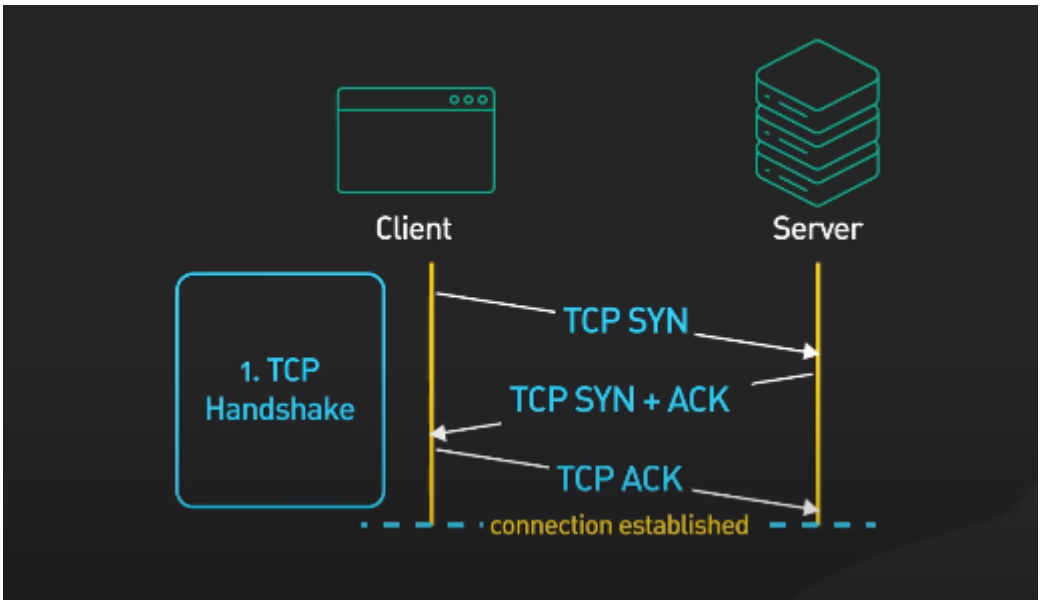
3. **Mã hóa dữ liệu**: Khi liên kết an toàn được thiết lập, mọi dữ liệu truyền qua VPN tunnel sẽ được mã hóa. Việc mã hóa có thể được thực hiện ở cấp độ gói tin(packet-level encryption) hoặc cấp độ kết nối (connection-level encryption), phụ thuộc vào giao thức và cài đặt cụ thể.

4. **Thay đổi khóa định kỳ**: Để tăng cường bảo mật, VPN thường thay đổi khóa mã hóa định kỳ theo thời gian hoặc dựa trên sự kiện cụ thể. Việc thay đổi khóa giúp ngăn chặn các tấn công và làm tăng khả năng chống lại các phương pháp giải mã lặp đi lặp lại.

5. **Xác thực**: VPN cũng có hể sử dụng các phương thức xác thực để đảm bảo tính xác định của các bên tham gia, việc xác thực có thể được thực hiện bằng cách sử dụng mật khẩu, chứng chỉ số, hoặc các phương tiện xác thực khác tùy thuộc vào cài đặt cụ thể của VPN.

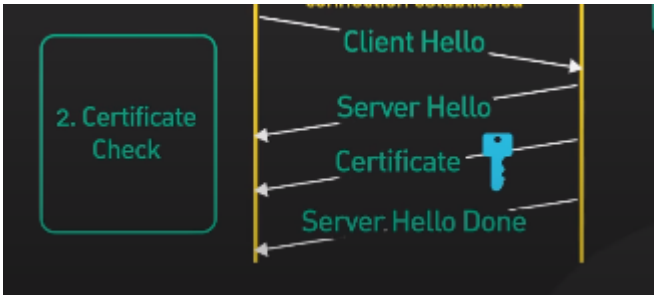
Lấy ví dụ về các bước handshake của TLS:

1. Đầu tiên Client và server thực hiện quá trình threeway-handshake để thiết lập kết nối TCP



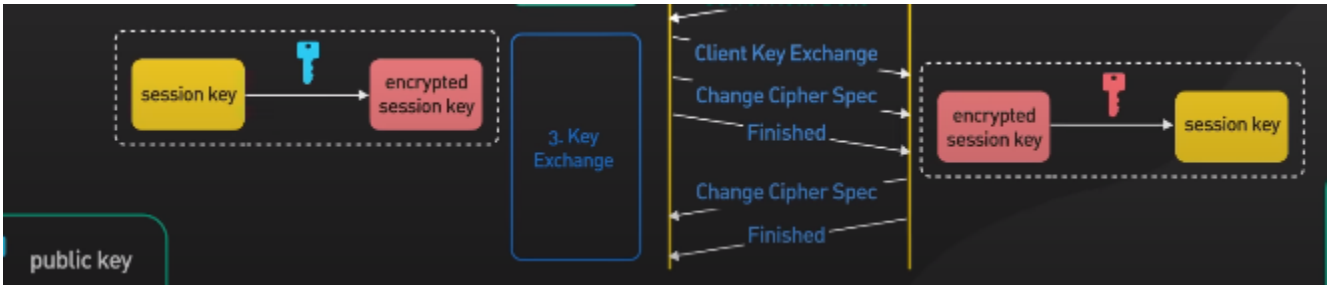
Hình 20: TCP handshake

2. Sau khi thực hiện handshake xong thì đây là bước bắt đầu thực hiện handshake của TLS. Client thực hiện gửi bản tin Hello đến server và muốn nói với server rằng: phiên bản TLS nào mà nó có thể hỗ trợ (TLS 1.2 hoặc TLS 1.3), và một danh sách các thuật toán mã hóa dữ liệu. Sau khi nhận được bản tin Hello từ client, server sẽ lựa chọn phiên bản TLS và thuật toán mã hóa dữ liệu dựa vào những options mà bản tin Hello đã được nhận và gửi lại bản tin Hello đến với client. Sau đó server sẽ gửi bản tin certificate đến cho client, bản tin certificate này bao gồm những thứ như là public key cho server,...Client sẽ sử dụng thuật toán mã hóa bất đối xứng (dữ liệu được mã hóa bằng public key chỉ có thể được giải mã bằng private key). Sau đó server sẽ gửi bản tin Hello done cho client nhằm xác nhận phiên bản TLS sử dụng, thuật toán mã hóa sử dụng và certificate được sử dụng và được đồng ý bởi cả 2 bên là server và client.



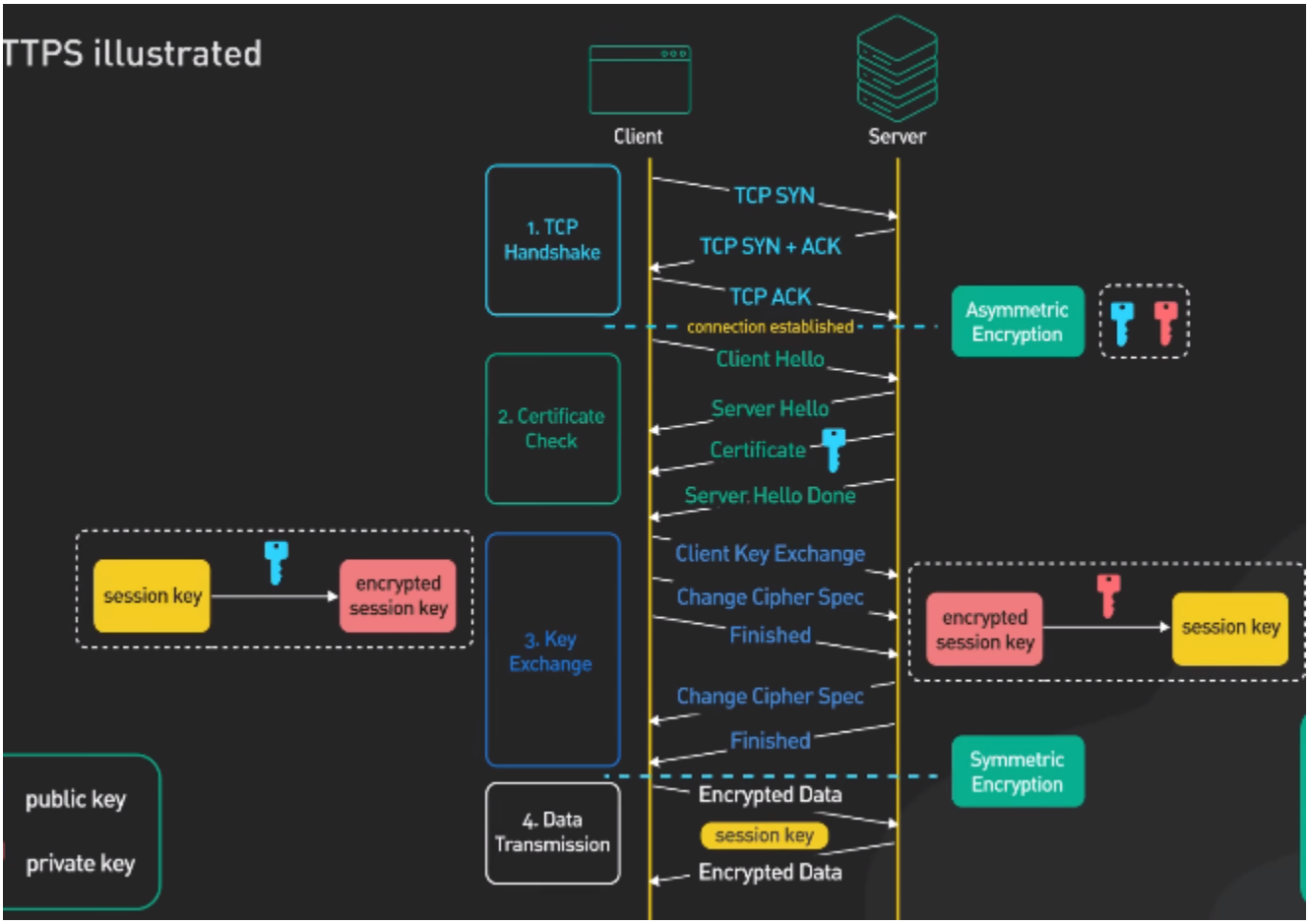
Hình 21: TLS handshake

3. Ở bước thứ ba là bước chia sẻ khóa để mã hóa dữ liệu giữa client và server. Client sẽ thực hiện gửi những bản tin Client Key exchange đến cho server, số lượng bản tin Key exchange phụ thuộc vào các tham số mà server đã gửi ở bước TLS handshake. Ở bước này, ta sẽ lấy ví dụ client sẽ sử dụng thuật toán RSA để mã hóa session key (được tạo bởi client) thành **encrypted session key** với public key đã được server gửi, và thực hiển gửi encrypted session key này cho server. Server nhận được encrypted session key này và dùng private key của mình để giải mã được session key. Đến bước này cả hai bên client và server đã đều có được session key.



Hình 22: Key exchange

4. Server và client dùng session key đã biết để mã hóa và tryền dữ liệu.

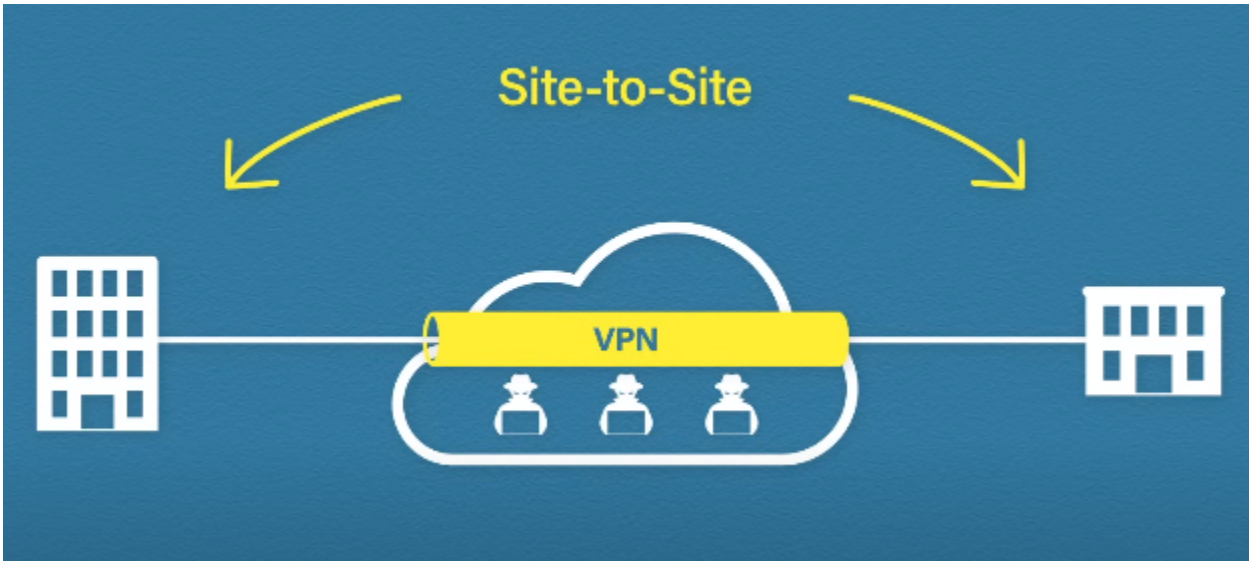


Hình 23: TLS session

2.3 Các loại VPN

2.3.1 Site-to-Site VPN

Site-to-site VPN là một loại kết nối VPN được thiết lập giữa hai hoặc nhiều vị trí vật lý khác nhau, chẳng hạn như hai chi nhành của một doanh nghiệp, hai văn phòng công ty, hoặc giữa một chi nhánh và một trung tâm dữ liệu. Kết nối này tạo ra một đường hầm an toàn qua mạng công cộng như internet, cho phép các mạng LAN ở các địa điểm khác nhau kết nối với nhau một cách an toàn và bảo mật. Site-to-site VPN được ưa chuộng trong các doanh nghiệp có nhiều chi nhánh hoặc văn phòng, giúp chúng kết nối với nhau một cách an toàn hiệu quả.

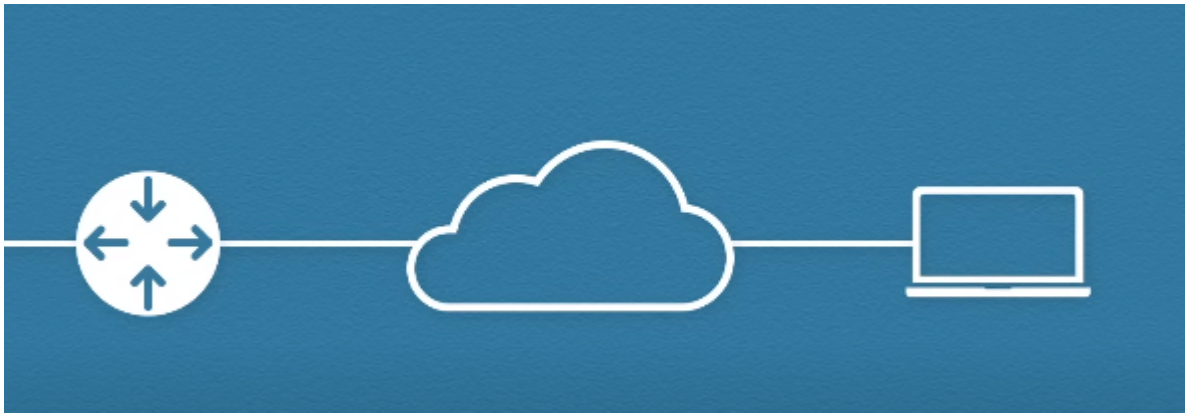


Hình 24: Site to site VPN

Site-to-site VPN cần phải được cấu hình trên cả hai mạng LAN (ở hai vị tri khác nhau) và được cấu hình ở router hoặc firewall ở cả 2. Một loại site to site VPN phổ biến được dùng đó là IPsec VPN. IPsec là một framework dược sử dụng để bảo mật phiên kết nối thông qua mạng IP. Mục đích của nó là bảo đảm sự toàn vẹn và xác thực của dữ liệu. Giả sử ta cần một VPN để gửi dữ liệu từ site A sang site B. Router sẽ xử lý dữ liệu này và nhận thấy rằng nó được dành cho site B và cần gửi dữ liệu này thông qua VPN. Gói tin IP được gửi sẽ được mã hóa với khóa đã được trao đổi trước đó. Sau khi mã hóa xong gói tin sẽ được đóng gói với VPN header và VPN trailer và thêm một IP header mới. Dữ liệu sẽ được truyền thông qua internet và đến site B. Gói tin được giải mã ở site B và chuyển tiếp dữ liệu đến đích. Site-to-site VPN luôn luôn được chạy ở cả hai phía.

2.3.2 Remote Access VPN

Remote Access VPN cho phép một thiết bị duy nhất (có thể là điện thoại, máy tính, tablet...) kết nối với một mạng LAN ở xa thông qua internet. Giống như site-to-site VPN, Remote Access VPN cho phép user từ xa gửi và nhận dữ liệu được mã hóa một cách an toàn qua mạng internet. Không giống như site-to-site VPN, remote access VPN yêu cầu một VPN client trên máy của user để kết nối với mạng LAN ở xa. Remote Access VPN thường sử dụng giao thức TLS để mã hóa và trao đổi dữ liệu.



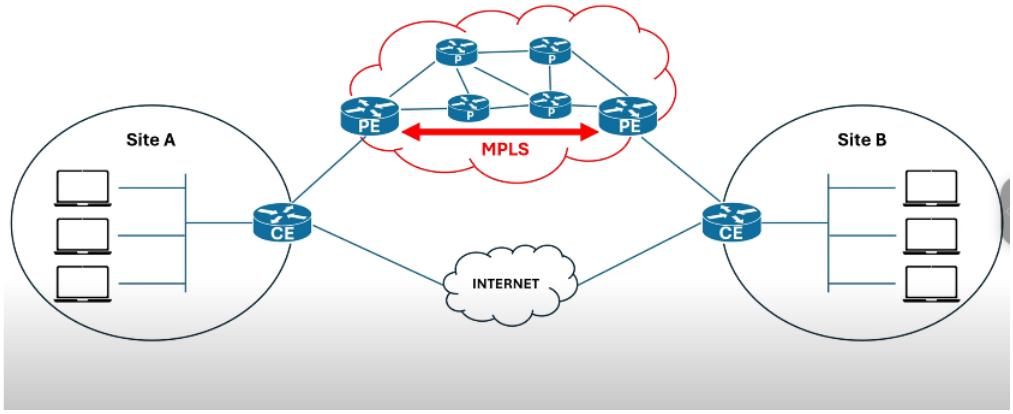
Hình 25: Remote Access VPN

Khi muốn thực hiện cấu hình VPN ta cần phải quyết định xem mình muốn sử dụng loại tunnel nào:

- Full-tunnel: Sau khi kết nối với VPN tất cả các traffic được phát ra từ user sẽ được chuyển tiếp đến mạng LAN mà thực hiện kết nối VPN.
- Split-tunnel: Chỉ chuyển tiếp lượng traffic dành cho mạng LAN mà thôi, tất cả các traffic khác đều được định tuyến như bình thường

2.3.3 MPLS VPN

Giống như site-to-site VPN, MPLS dùng để kết nối 2 hoặc nhiều mạng LAN ở vị trí xa nhau. Tuy nhiên khác với site-to-site VPN sử dụng tunnel thông qua internet thì MPLS VPN lại không sử dụng tunnel thông qua internet mà 'tunnel' này được cung cấp ở phía ISP.



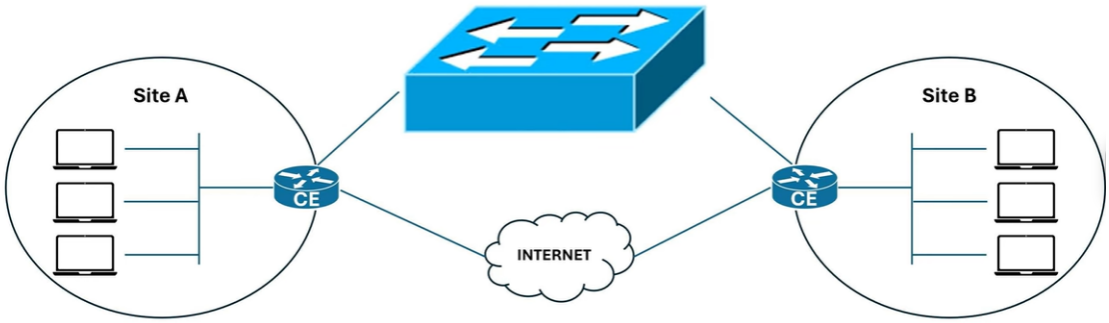
Hình 26: MPLS VPN

Ở trong hình site A và site B sẽ được kết nối với một router PE (Provider Edge) của bên phía ISP. Một router PE có thể được phục vụ cho nhiều site. Kênh truyền VPN sẽ được thiết lập giữa 2 router PE với nhau thông qua công nghệ MPLS. Việc sử dụng MPLS VPN sẽ khiến cho kết nối được nhanh và ổn định nhưng lại phải trả giá bởi chi phí cao.MPLS VPN được chia làm 2 loại:

- MPLS Layer 2 VPN

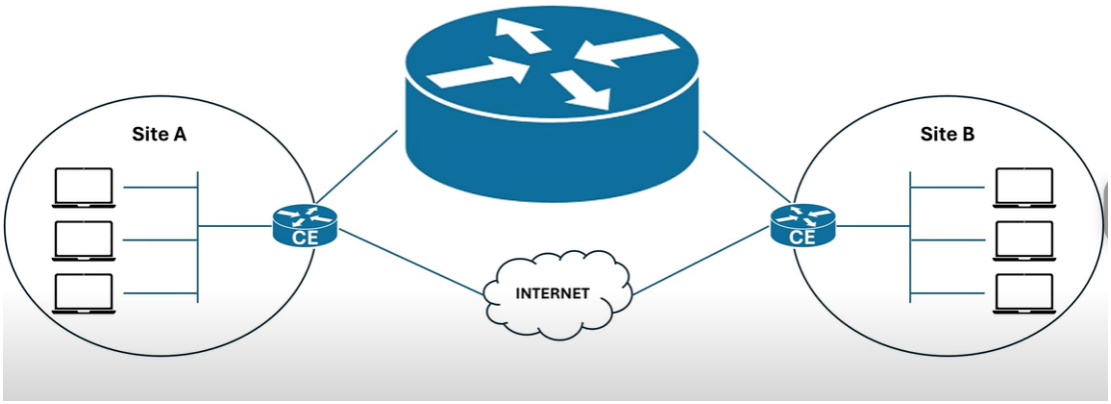
- MPLS Layer 3 VPN

MPLS Layer 2 VPN thì ở đây ta có thể coi phần được kết nối với nhà mạng (ISP) như là kết nối với một con switch và ở trên 2 router nối 2 site có interface được kết nối MPLS VPN sẽ có địa chỉ IP cùng một subnet và như vậy 2 site có thể kết nối với nhau.



Hình 27: MPLS Layer 2 VPN

MPLS Layer 3 VPN thì ở đâu ta có thể coi phần được kết nối với nhà mạng như là một con router và ta cần phải thống nhất giữa 2 site về địa chỉ IP cũng như giao thức định tuyến.



Hình 28: MPLS Layer 3 VPN

MPLS: MPLS (Multiprotocol Label Switching) là một công nghệ trong lĩnh vực mạng máy tính được thiết kế để cải thiện hiệu suất và quản lý mạng. Nó là một giao thức chuyển mạng mà không phụ thuộc vào các giao thức định tuyến cổ điển như IP (Internet Protocol). MPLS sử dụng các nhãn (labels) để định danh và chuyển tiếp gói dữ liệu trong mạng. Nhãn này được thêm vào gói dữ liệu tại điểm xuất phát và được sử dụng để xác định con đường chuyển tiếp qua mạng cho đến khi đến đích. Các router MPLS sử dụng thông tin trong nhãn để đưa ra quyết định về con đường chuyển tiếp, giảm bớt sự phụ thuộc vào quy trình định tuyến truyền thống. MPLS được sử dụng rộng rãi trong các mạng lớn, đặc biệt là trong các mạng của các nhà cung cấp dịch vụ internet (ISP) và mạng doanh nghiệp. Nó mang lại khả năng chuyển mạng hiệu quả và linh hoạt, giúp cải thiện hiệu suất, quản lý và chất lượng dịch vụ trong môi trường mạng phức tạp.

2.4 OpenVPN lab

OpenVPN là một ứng dụng để triển khai VPN. OpenVPN là một phần mềm open-source VPN có khả năng cung cấp một kênh kết nối bảo mật thông qua interne. Nó cho phép người dùng kết nối tới mạng nội bộ từ một địa điểm ở xa, tạo một tunnel ảo cho việc truyền dữ liệu. OpenVPN sử dụng SSL/TLS như một giao thức để bảo mật cho việc trao đổi khóa đảm bảo được tính xác thức và sự toàn vẹn của dữ liệu. Ở trong môi trường lab này, em sẽ thực hành cấu hình VPN cho server và client có thể giao tiếp được với nhau.
Đầu tiền chạy script sau để tải về file bash script tự động thiết lập key, certificate trên server:

wget https://git.io/vpn -O openvpn-install.sh && bash openvpn-install.sh

Sau khi tải về và chạy file bash xong tiến hành, điền thông tin để thiết lập cấu hình cho kết nối VPN. Khi đã điền thông tin xong thực hiện mở port cho openvpn:

```
root@buidung:~# ufw status
Status: active

To                         Action      From
--                         ------      ----
123/udp                    ALLOW       Anywhere
1194/udp                   ALLOW       Anywhere
22/tcp                     ALLOW       Anywhere
123/udp (v6)               ALLOW       Anywhere (v6)
1194/udp (v6)              ALLOW       Anywhere (v6)
22/tcp (v6)                ALLOW       Anywhere (v6)
```

Hình 29: Firewall status

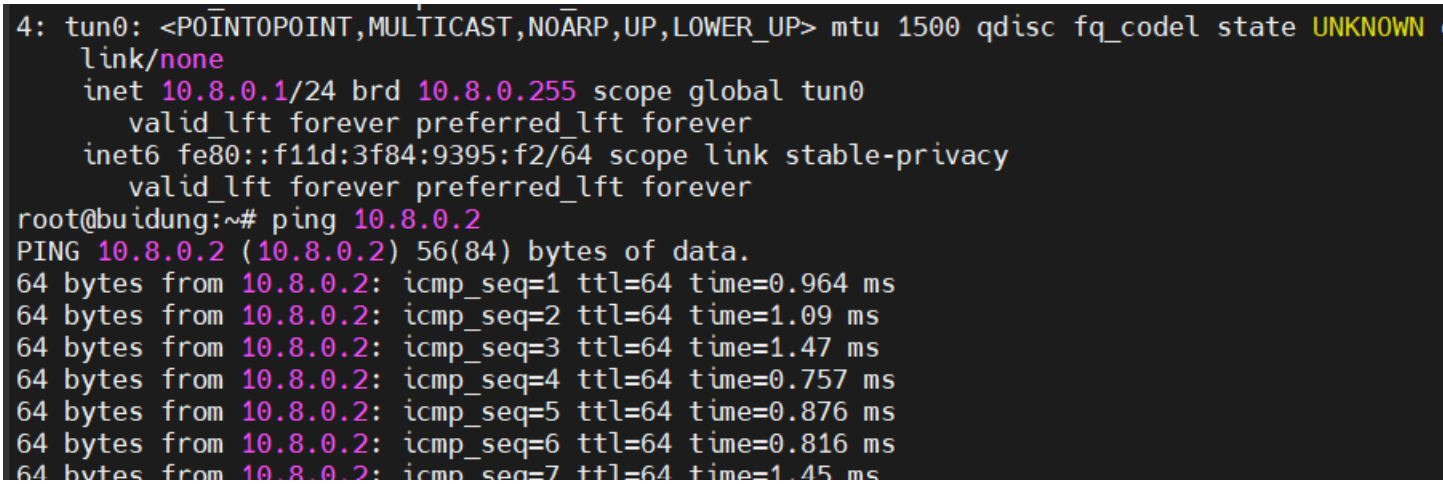
Sau khi đã mở port xong thực hiện scp file *openvpn-client.ovpn* đến cho client bằng câu lệnh. **Lưu ý** trước khi thực hiện scp thì ta phải copy file trên từ thư mục home của root về thư mục home của user:

scp buidung@192.168.74.136:/home/buidung/openvpn-client.ovpn .

Sau khi đã scp xong thực hiện câu lệnh *sudo openvpn –config openvpn-client.ovpn* để chạy file config trên client. Sau đó mở tab mới lên và thu được kết quả:

```
6: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN
    link/none
    inet 10.8.0.2/24 brd 10.8.0.255 scope global tun0
       valid_lft forever preferred_lft forever
    inet6 fe80::23f4:f194:1a4d:551c/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
buidung@buidung:~$ ping 10.8.0.1
PING 10.8.0.1 (10.8.0.1) 56(84) bytes of data.
64 bytes from 10.8.0.1: icmp_seq=1 ttl=64 time=1.38 ms
64 bytes from 10.8.0.1: icmp_seq=2 ttl=64 time=1.25 ms
^C
--- 10.8.0.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 1.251/1.315/1.380/0.064 ms
```

Hình 30: Tunnel trên client và test ping



Hình 31: Tunnel trên server và test ping

Như vậy em đã thành công tạo một kết nối VPN sử dụng OpenVPN.