VIET NAM NATIONAL UNIVERSITY HCMC
UNIVERSITY OF INFORMATION TECHNOLOGY

PROJECT REPORT

SUBJECT: NETWORK AND SYSTEM ADMINISTARTION

Project: OpenVPN

LECTURER: Trần Thị Dung

Members:

Dương Phan Hiếu Nghĩa	21521179
Nguyễn Đức Hoàng	21520869
Khấu Đặng Tường Minh	21522337

OO Ho Chi Minh City, 2023 OO

Lecture's Comments

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Date

Lecturer

Table of Contents

I. INTRODUCTION

1.1. General Information

VPN Definition:

A virtual private network provides online privacy and anonymity by creating a private network from a public internet connection.

VPN services establish secure and encrypted connections to provide greater privacy.

VPN can be divided into two main types:

Remote access VPN allows individual users to connect to a private network from a remote location, providing secure access to resources such as files or applications.

Site-to-site VPN establishes secure connections between different physical locations or networks, allowing smooth communication and resource sharing among these locations.

OpenVPN:

OpenVPN is a robust and highly flexible VPN daemon. OpenVPN supports SSL/TLS security, ethernet bridging, TCP or UDP tunnel transport through proxies or NAT, support for dynamic IP addresses and DHCP, scalability to hundreds or thousands of users, and portability to most major OS platforms.

OpenVPN supports conventional encryption using a pre-shared secret key (Static Key mode) or public key security (SSL/TLS mode) using client & server certificates. OpenVPN also supports non-encrypted TCP/UDP tunnels.

OpenVPN is designed to work with the TUN/TAP virtual networking interface that exists on most platforms.

Overall, OpenVPN aims to offer many of the key features of IPSec but with a relatively lightweight footprint.

1.2. Components

OpenVPN Server:

The OpenVPN server is responsible for accepting incoming VPN connections and handling the encryption and decryption of data. It runs on a server machine and manages the VPN connections from clients.

OpenVPN Client:

The OpenVPN client runs on the devices that want to establish a VPN connection with the server. It can be installed on various devices such as computers, smartphones, or routers. The client is responsible for initiating the connection to the server and encrypting and decrypting the data sent over the VPN.

OpenVPN Protocol:

OpenVPN uses a strong encryption and authentication protocol to create a secure VPN connection. Typically, OpenVPN uses UDP or TCP protocols to transmit data over the network. By default, OpenVPN uses port 1194, but it can be set to a custom port if needed.

Encryption and Authentication:

OpenVPN uses strong encryption algorithms to secure the data transmitted between the client and the server. It employs techniques such as symmetric key encryption, public-key cryptography, and digital certificates. Authentication mechanisms, such as

username/password or digital certificates, are used to verify the identities of the client and server.

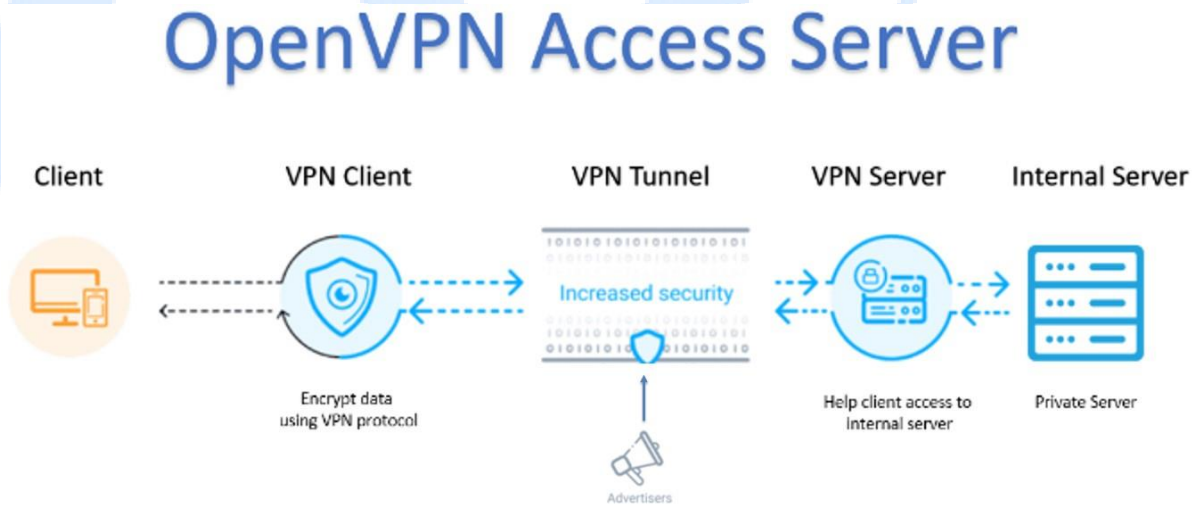
Virtual IP Address:

When a client successfully connects to the OpenVPN server, it is assigned a virtual IP address within a private subnet. This virtual IP address allows the client to communicate with resources on the server's network as if it were physically present on that network.

Configuration and Management:

The OpenVPN server and client require configuration files that specify various parameters such as server address, port number, encryption settings, authentication methods, and more. These configuration files are used to set up and manage the VPN connections.

1.3. Operation



Step 1: User Authentication

Users connecting to OpenVPN Access Server are authenticated through a variety of methods, including username/password, client certificates, or multifactor authentication.

Authentication ensures that only authorized users can establish a secure connection with the server.

Step 2: TLS Handshake

The TLS (Transport Layer Security) handshake is a crucial part of the connection process in OpenVPN Access Server.

During the TLS handshake, the server and client authenticate each other, exchange cryptographic information, and agree on encryption algorithms.

Step 3: Key Exchange

The TLS handshake involves key exchange, where both parties agree on a set of encryption keys to secure communication.

OpenVPN Access Server often utilizes the Diffie-Hellman key exchange algorithm for secure key negotiation.

Step 4: Encryption of Data

Once the TLS handshake is complete, OpenVPN Access Server establishes a secure tunnel with a key exchange.

Data transmitted between the client and the server is encrypted using robust encryption algorithms, with AES-256-GCM commonly employed as the default.

Step 5: Data Transmission through OpenVPN Tunnel

The encrypted message traverses the established OpenVPN tunnel, leveraging SSL and TLS protocols.

SSL/TLS protocols maintain an encrypted connection between the user's device and the OpenVPN Access Server, enhancing the overall security of the data in transit.

Step 6: Forwarding to the Internal Website

OpenVPN Access Server receives the encrypted message and securely forwards it to the intended website.

The data forwarded to the website remains encrypted during transmission, providing end-to-end security.

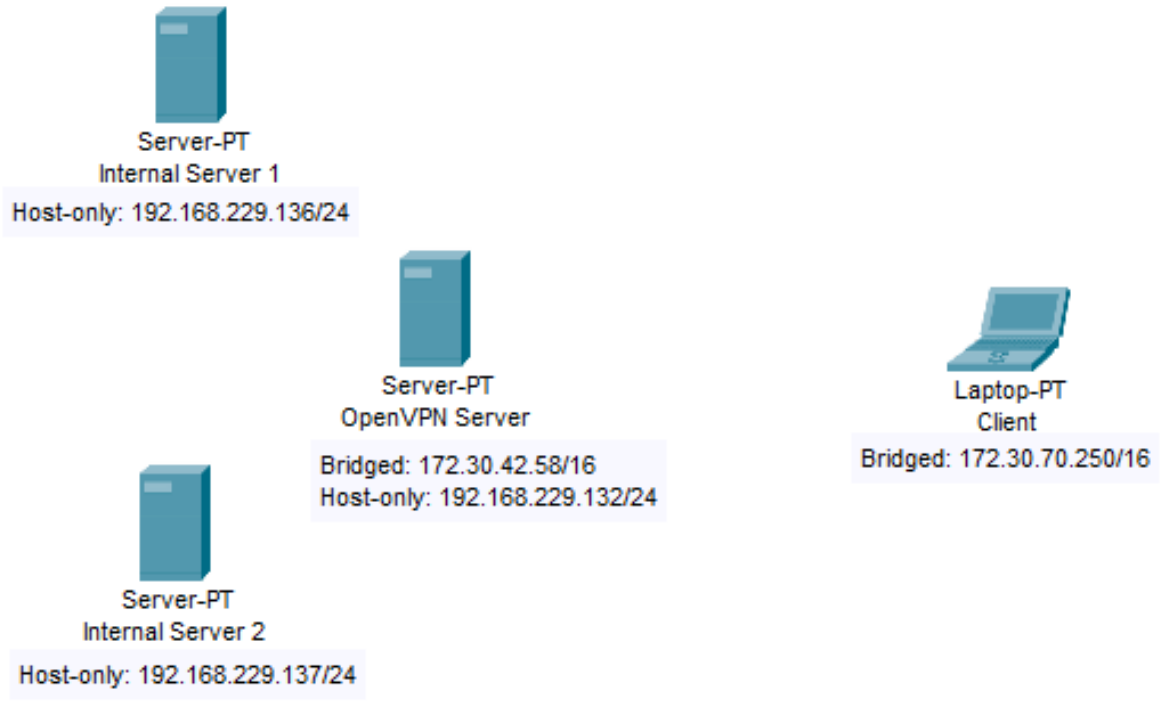
Step 7: Response Path

When the website responds, the response follows a similar path in reverse through the encrypted tunnel established between the OpenVPN Access Server and the user.

This ensures that the response is also encrypted, maintaining the integrity and confidentiality of the transmitted data.

II. IMPLEMENTATION

2.1. Topology



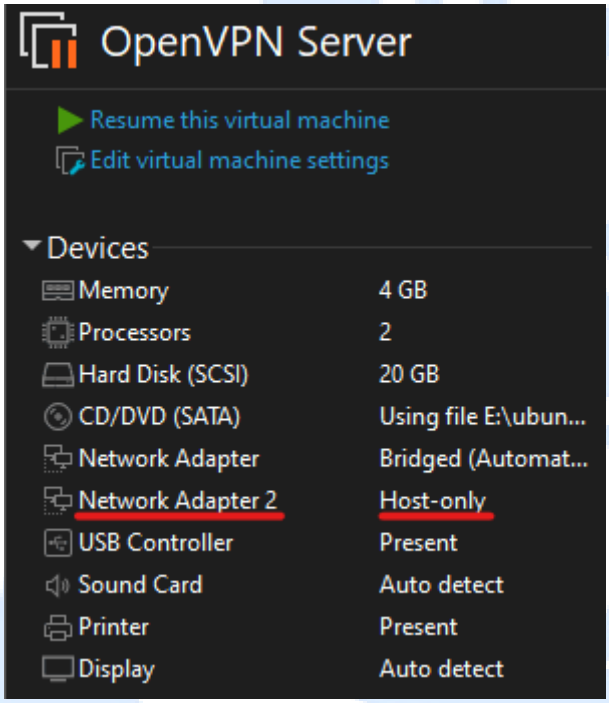
Device	IP Address	Installation
OpenVPN Server	Bridged: 172.30.42.58/16 Host-only: 192.168.229.132/24	Ubuntu 22.04.3 LTS ca-certificates net-tools OpenVPN Access Server 2.12.2 OpenSSL 3.0.2
Internal Server 1	Host-only: 192.168.229.136/24	Ubuntu 22.04.3 LTS apache/2.4.57 (Debian) openssh-server 9.3 openssl 3.10.10
Internal Server 2	Host-only: 192.169.229.137/24	Ubuntu 22.04.3 LTS apache/2.4.57 (Debian) openssh-server 9.3 openssl 3.10.10

Client	Bridged: 172.30.70.250/16	Kali GNU/Linux Rolling 2023.3 openssh-client 9.4 openssl 3.0.11 openvpn 2.6.3

2.2. Installation

2.2.1. OpenVPN Server

Operation system and NIC:



```
nghia@nghia-virtual-machine:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.30.42.58  netmask 255.255.0.0  broadcast 172.30.255.255
        inet6 fe80::1195:ebcd:7ce5:848  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:5e:3a:17  txqueuelen 1000  (Ethernet)
        RX packets 1121  bytes 221300 (221.3 KB)
        RX errors 0  dropped 5  overruns 0  frame 0
        TX packets 55  bytes 7411 (7.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 19  base 0x2000

ens34: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.229.132  netmask 255.255.255.0  broadcast 192.168.229.255
        inet6 fe80::a83c:e238:9561:e681  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:5e:3a:21  txqueuelen 1000  (Ethernet)
        RX packets 4  bytes 654 (654.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 45  bytes 6153 (6.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 16  base 0x2080

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 1259  bytes 94759 (94.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1259  bytes 94759 (94.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Install ca-certificates:

```
nghia@nghia-virtual-machine:~$ sudo apt-get install ca-certificates -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20230311ubuntu0.22.04.1).
```

Install net-tools:

```
nghia@nghia-virtual-machine:~$ sudo apt-get install net-tools
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
net-tools is already the newest version (1.60+git20181103.0eebece-1ubuntu5).
```

Install openvpn-as:

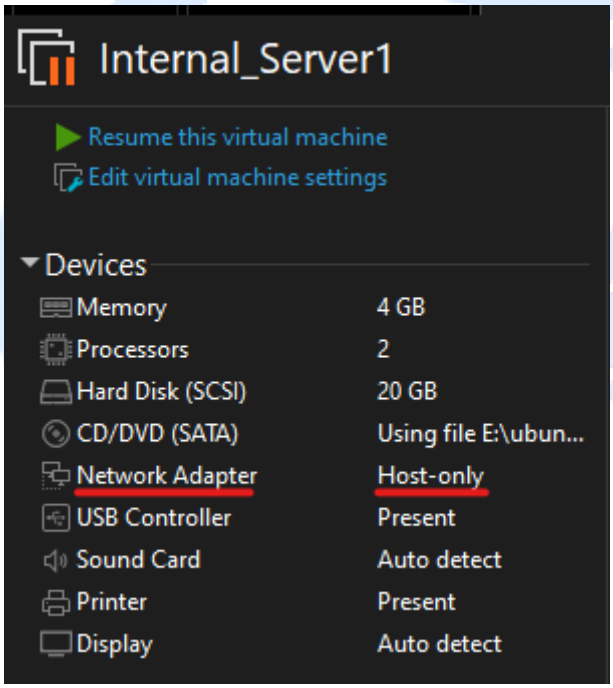
```
nghia@nghia-virtual-machine:~$ sudo apt-get install openvpn-as
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages will be upgraded:
  openvpn-as
1 upgraded, 0 newly installed, 0 to remove and 46 not upgraded.
```

Install openssl:

```
nghia@nghia-virtual-machine:~$ sudo apt-get install openssl -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssl is already the newest version (3.0.2-0ubuntu1.12).
openssl set to manually installed.
```

2.2.2. Internal Server 1

Operation system and NIC:



```
nghia@nghia-virtual-machine:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.229.136  netmask 255.255.255.0  broadcast 192.168.229.255
        inet6 fe80::be11:9492:81e6:d80f  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:47:1c:73  txqueuelen 1000  (Ethernet)
        RX packets 25  bytes 3114 (3.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 132  bytes 17291 (17.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 19  base 0x2000

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 440  bytes 34943 (34.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 440  bytes 34943 (34.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Install apache2:

```
nghia@nghia-virtual-machine:~$ sudo apt-get install apache2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.52-1ubuntu4.7).
The following packages were automatically installed and are no longer required:
  linux-headers-6.2.0-26-generic linux-hwe-6.2-headers-6.2.0-26 linux-image-6.2.0-26-generic
  linux-modules-6.2.0-26-generic linux-modules-extra-6.2.0-26-generic
```

Check apache2 status:

```
nghia@nghia-virtual-machine:~$ service apache2 status
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2023-12-05 09:37:23 +07; 4min 0s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 921 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 977 (apache2)
      Tasks: 55 (limit: 4556)
     Memory: 7.4M
        CPU: 88ms
     CGroup: /system.slice/apache2.service
             ├─977 /usr/sbin/apache2 -k start
             ├─979 /usr/sbin/apache2 -k start
             └─980 /usr/sbin/apache2 -k start
```

Install openssh-server:

```
nghia@nghia-virtual-machine:~$ sudo apt-get install openssh-server -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-server is already the newest version (1:8.9p1-3ubuntu0.4).
The following packages were automatically installed and are no longer required:
  linux-headers-6.2.0-26-generic linux-hwe-6.2-headers-6.2.0-26 linux-image-6.2.0-26-generic
  linux-modules-6.2.0-26-generic linux-modules-extra-6.2.0-26-generic
```

Check openssh-server status:

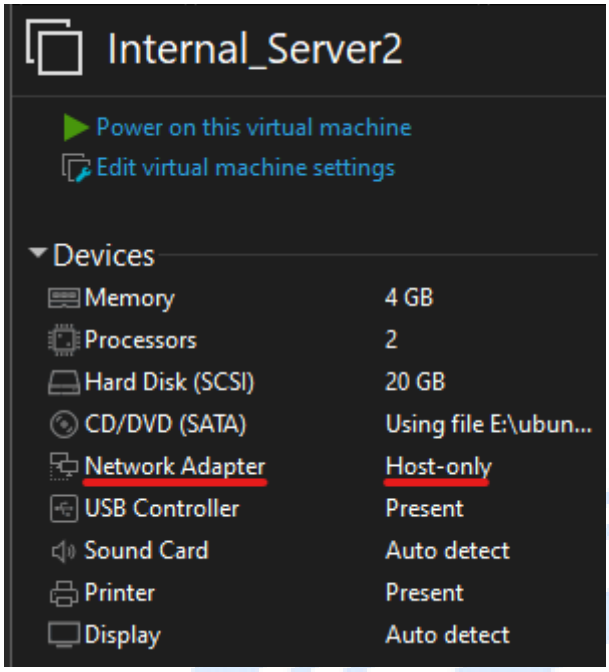
```
nghia@nghia-virtual-machine:~$ service ssh status
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2023-12-05 09:37:23 +07; 5min ago
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 932 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 967 (sshd)
      Tasks: 1 (limit: 4556)
     Memory: 3.0M
        CPU: 39ms
     CGroup: /system.slice/ssh.service
             └─967 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
```

Install openssl:

```
nghia@nghia-virtual-machine:~$ sudo apt-get install openssl -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssl is already the newest version (3.0.2-0ubuntu1.12).
openssl set to manually installed.
The following packages were automatically installed and are no longer required:
  linux-headers-6.2.0-26-generic linux-hwe-6.2-headers-6.2.0-26 linux-image-6.2.0-26-generic
  linux-modules-6.2.0-26-generic linux-modules-extra-6.2.0-26-generic
```

2.2.3. Internal Server 2

Operation system and NIC:




```
nghia@nghia-virtual-machine:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.229.137  netmask 255.255.255.0  broadcast 192.168.229.255
        inet6 fe80::20c:29ff:fe03:8bde  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:03:8b:de  txqueuelen 1000  (Ethernet)
        RX packets 5  bytes 931 (931.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 31  bytes 4502 (4.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 19  base 0x2000

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 719  bytes 53447 (53.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 719  bytes 53447 (53.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Install apache2:

```
nghia@nghia-virtual-machine:~$ sudo apt-get install apache2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.52-1ubuntu4.7).
```

Check apache2 status:

```
nghia@nghia-virtual-machine:~$ service apache2 status
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2023-12-05 09:46:17 +07; 3min 7s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 1227 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 1355 (apache2)
      Tasks: 55 (limit: 4556)
     Memory: 7.4M
        CPU: 93ms
     CGroup: /system.slice/apache2.service
             ├─1355 /usr/sbin/apache2 -k start
             ├─1356 /usr/sbin/apache2 -k start
             └─1357 /usr/sbin/apache2 -k start
```

Install openssh-server:

```
nghia@nghia-virtual-machine:~$ sudo apt-get install openssh-server -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-server is already the newest version (1:8.9p1-3ubuntu0.4).
```

Check openssh-server status:

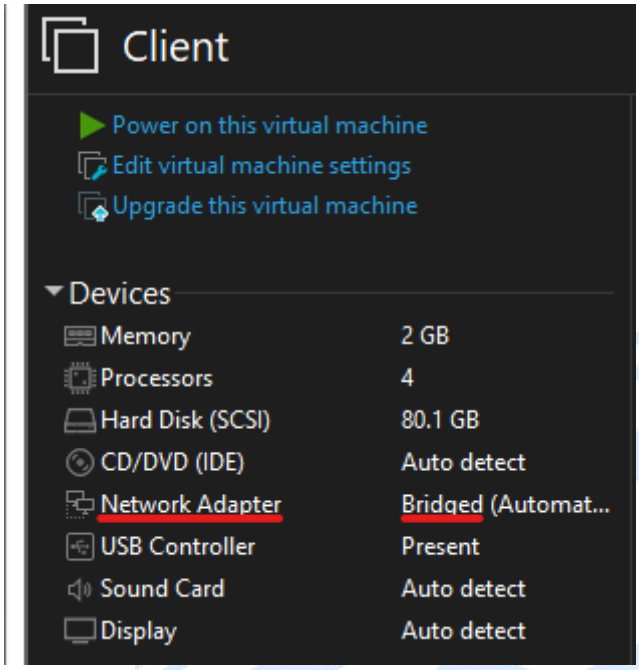
```
nghia@nghia-virtual-machine:~$ service ssh status
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2023-12-05 09:46:15 +07; 4min 33s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 952 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 996 (sshd)
      Tasks: 1 (limit: 4556)
     Memory: 3.0M
        CPU: 44ms
     CGroup: /system.slice/ssh.service
             └─996 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
```

Install openssl:

```
nghia@nghia-virtual-machine:~$ sudo apt-get install openssl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssl is already the newest version (3.0.2-0ubuntu1.12).
openssl set to manually installed.
```

2.2.4. Client

Operating system and NIC:




```
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.30.70.250  netmask 255.255.0.0  broadcast 172.30.255.255
        inet6 fe80::48e3:9244:6726:c402  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:4d:96:48  txqueuelen 1000  (Ethernet)
        RX packets 3286  bytes 562757 (549.5 KiB)
        RX errors 0  dropped 24  overruns 0  frame 0
        TX packets 18  bytes 2056 (2.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 4  bytes 240 (240.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4  bytes 240 (240.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Install openssh-client:

```
┌──(kali㉿kali)-[~]
└─$ sudo apt-get install openssh-client -y
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following additional packages will be installed:
  openssh-server openssh-sftp-server
Suggested packages:
  keychain libpam-ssh monkeysphere ssh-askpass molly-guard ufw
The following packages will be upgraded:
  openssh-client openssh-server openssh-sftp-server
```

Install openssl:

```
┌──(kali㉿kali)-[~]
└─$ sudo apt-get install openssl -y
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following additional packages will be installed:
  libssl3
The following packages will be upgraded:
  libssl3 openssl
```

Install openvpn:

```
┌──(kali㉿kali)-[~]
└─$ sudo apt-get install openvpn -y
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
Suggested packages:
  resolvconf openvpn-dco-dkms openvpn-systemd-resolved
The following packages will be upgraded:
  openvpn
```

2.3. Configuration

2.3.1. OpenVPN Server

Set up openvpn-as:

```
nghia@nghia-virtual-machine:~$ sudo /usr/local/openvpn_as/bin/ovpn-init --force

          OpenVPN Access Server
          Initial Configuration Tool
-------------------------------------------------------
OpenVPN Access Server End User License Agreement (OpenVPN-AS EULA)

    1. Copyright Notice: OpenVPN Access Server License;
       Copyright (c) 2009-2023 OpenVPN Inc. All rights reserved.
       "OpenVPN" is a trademark of OpenVPN Inc.
    2. Redistribution of OpenVPN Access Server binary forms and related documents,
       are permitted provided that redistributions of OpenVPN Access Server binary
       forms and related documents reproduce the above copyright notice as well as
       a complete copy of this EULA.
    3. You agree not to reverse engineer, decompile, disassemble, modify,
       translate, make any attempt to discover the source code of this software,
       or create derivative works from this software.
    4. The OpenVPN Access Server is bundled with other open source software
       components, some of which fall under different licenses. By using OpenVPN
       or any of the bundled components, you agree to be bound by the conditions
       of the license for each respective component. For more information, you can
       find our complete EULA (End-User License Agreement) on our website
```

Accept the license:

```
   11. Purchasing an activation key does not entitle you to any special rights or
       privileges, except the ones explicitly outlined in this user agreement.
       Unless otherwise arranged prior to your purchase with OpenVPN Inc.,
       software maintenance costs and terms are subject to change after your
       initial purchase without notice. In case of price decreases or special
       promotions, OpenVPN Inc. will not retrospectively apply credits or price
       adjustments toward any licenses that have already been issued. Furthermore,
       no discounts will be given for license maintenance renewals unless this is
       specified in your contract with OpenVPN Inc.

Please enter 'yes' to indicate your agreement [no]: yes
```

Set primary access server node:

```
Will this be the primary Access Server node?
(enter 'no' to configure as a backup or standby node)
> Press ENTER for default [yes]: yes
```

Chose interface to be used by the Admin Web UI:

```
Please specify the network interface and IP address to be
used by the Admin Web UI:
(1) all interfaces: 0.0.0.0
(2) ens33: 172.30.42.58
(3) ens34: 192.168.229.132
Please enter the option number from the list above (1- 3).
> Press Enter for default [1]: 1
```

Algorithms use for OpenVPN CA: secp384r1

```
What public/private type/algorithms do you want to use for the OpenVPN CA?

Recommended choices:

rsa       - maximum compatibility
secp384r1 - elliptic curve, higher security than rsa, allows faster connection setup and smaller user
 profile files
showall   - shows all options including non-recommended algorithms.
> Press ENTER for default [secp384r1]:secp384r1
```

Algorithms use for Web-Certificates: secp 384r1

```
What public/private type/algorithms do you want to use for the self-signed web certificate?

Recommended choices:

rsa       - maximum compatibility
secp384r1 - elliptic curve, higher security than rsa, allows faster connection setup and smaller user
 profile files
showall   - shows all options including non-recommended algorithms.
> Press ENTER for default [secp384r1]:secp384r1
```

Port number for Admin Web UI: 943

```
Please specify the port number for the Admin Web UI.
> Press ENTER for default [943]: 943
```

Port number for OpenVPN Daemon: 443

```
Please specify the TCP port number for the OpenVPN Daemon
> Press ENTER for default [443]: 443
```

Accept traffic be routed through VPN:

```
Should client traffic be routed by default through the VPN?
> Press ENTER for default [yes]: yes
```

Accept DNS traffic be routed through VPN:

```
Should client DNS traffic be routed by default through the VPN?
> Press ENTER for default [yes]: yes
```

Restrict access to private subnets → Access Control

```
Private subnets detected: ['192.168.229.0/24', '172.30.0.0/16']

Should private subnets be accessible to clients by default?
> Press ENTER for default [yes]: no
```

Create admin account:

```
> Specify the username for an existing user or for the new user account: group4
Type a password for the 'group4' account (if left blank, a random password will be generated):
Confirm the password for the 'group4' account:
```

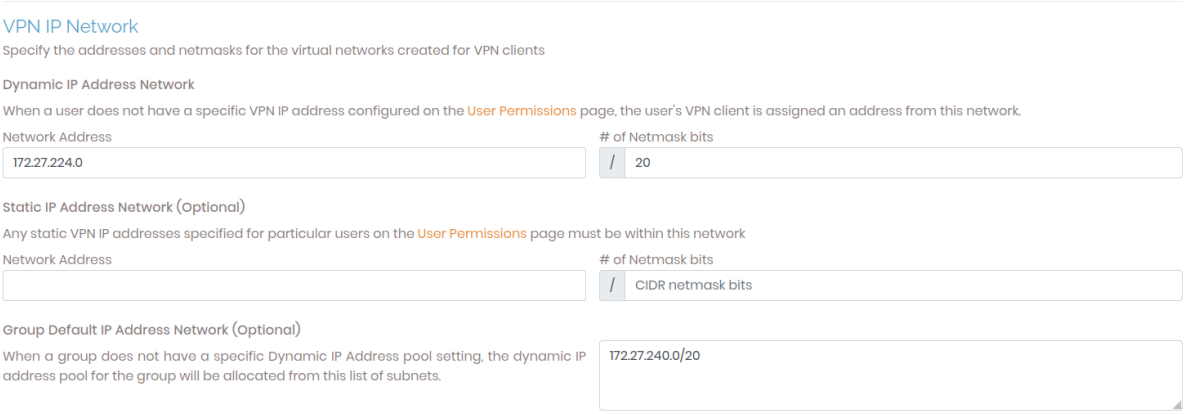
Active the account:

```
> Please specify your Activation key (or leave blank to specify later): ewogICJub25jZSIgOiAiNEUxQTFEN
DMwMzMyNEQ3QTlEQUZEQjUxMjUyODlGNTg5RTY2ODExN0FGQzVGNjhDNjE2MTZCMzVEREIzMEY3MyIsCiAgInN1YmtleSIgOiAiQV
NVdXdKblV1RlBGaHR3YnVZSU9ua21fQVNiUFRlWkRnVXJlWmRHUUVvbEVhSUlhTGRaZlJnUW5fNTU0OWQ1ZjcwZTBjMTc3ODVjM2E
4ZmZkZGNjN2JiMmQxZDNmY2Y4YiIsCiAgImlhdCIgOiAxNjk5NjkwMzU4Cn0=
Activation succeeded
```

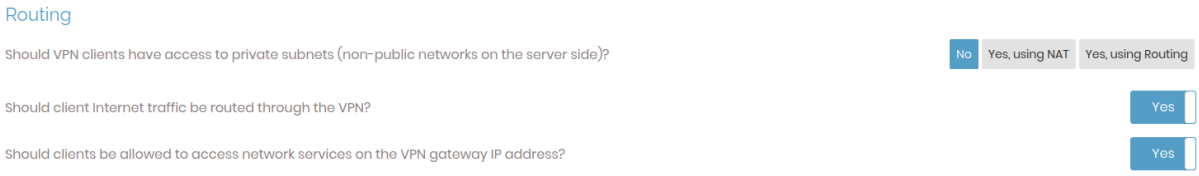
Initial configuration complete:

```
Initial Configuration Complete!

You can now continue configuring OpenVPN Access Server by
directing your Web browser to this URL:

https://172.30.42.58:943/admin

During normal operation, OpenVPN AS can be accessed via these URLs:
Admin  UI: https://172.30.42.58:943/admin
Client UI: https://172.30.42.58:943/
```

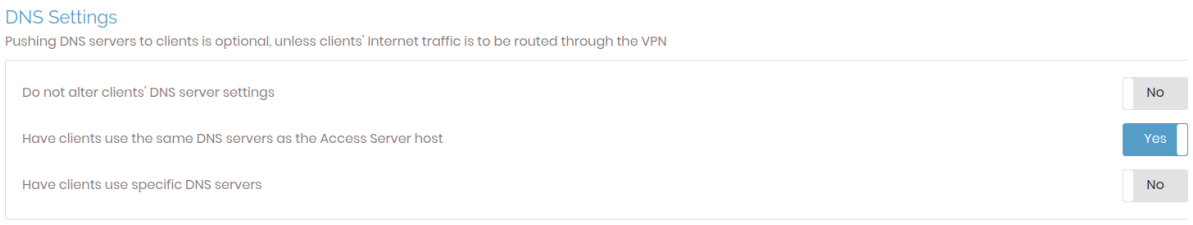
Config VPN IP Network:



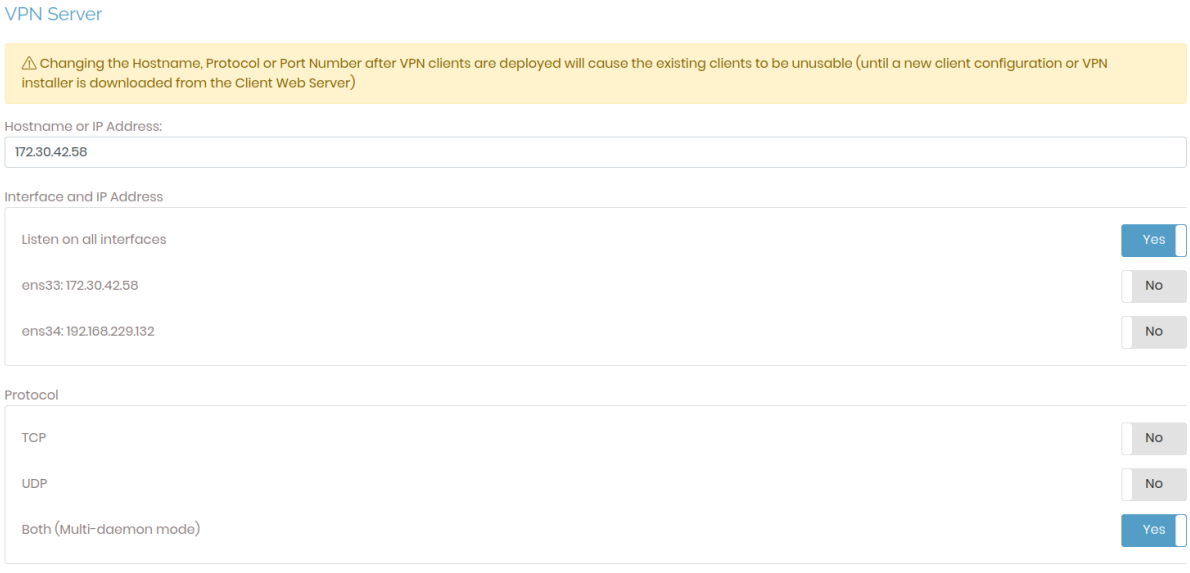
Config Routing:



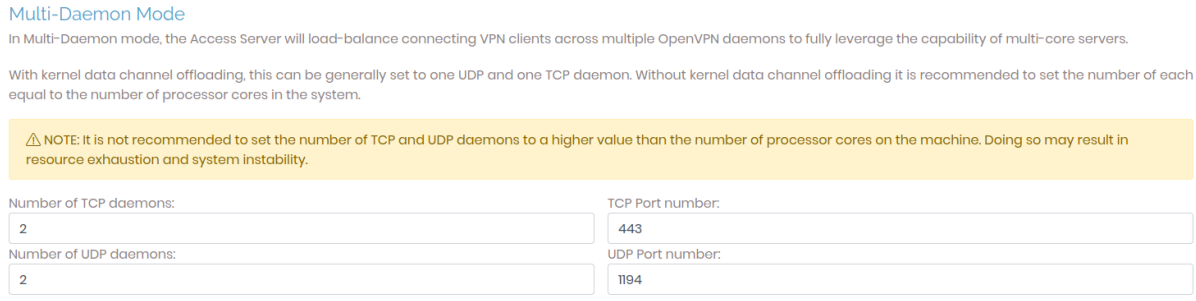
Config DNS Setting:



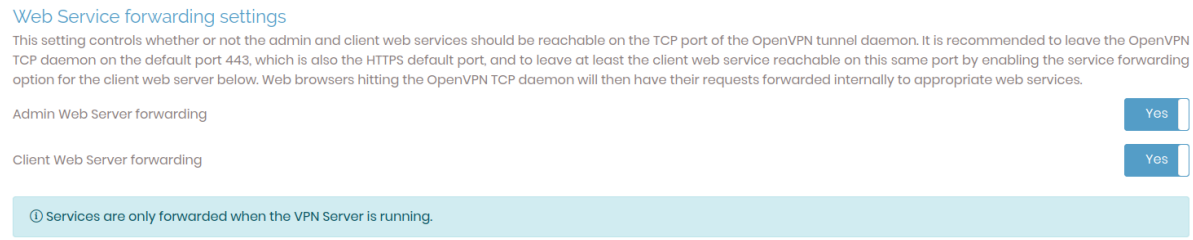
Config VPN Server:



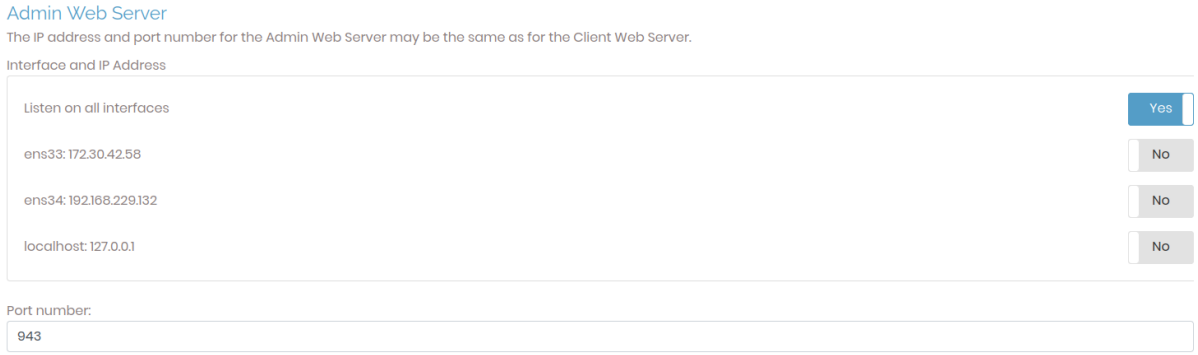
Config Multi-Daemon Mode:



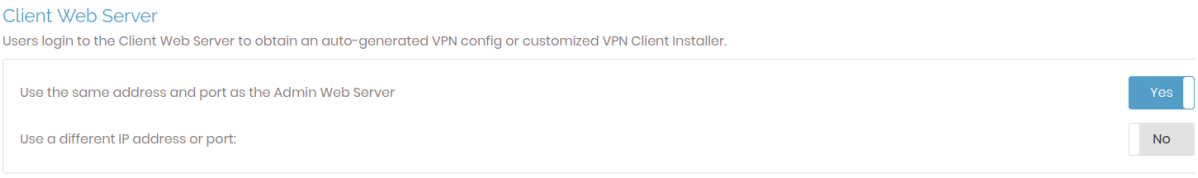
Config Web Service forwarding settings:



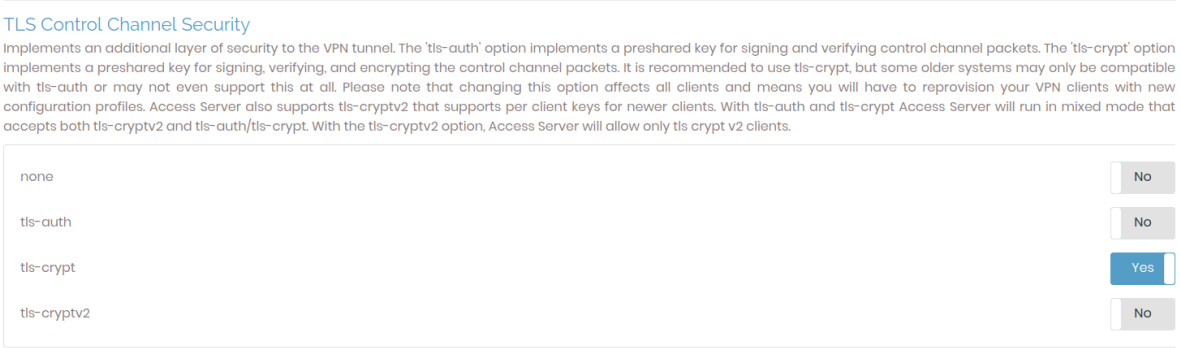
Config Admin Web Server:



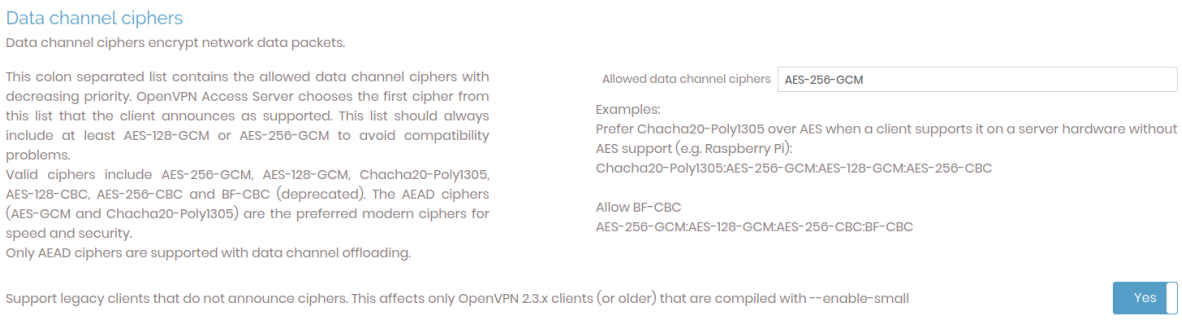
Config Client Web Server:



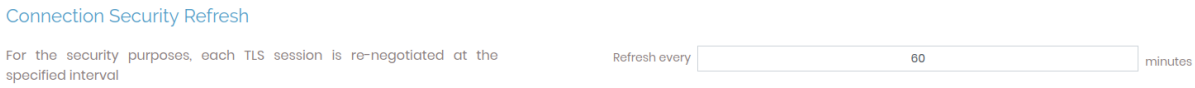
Config TLS Control Channel Security:



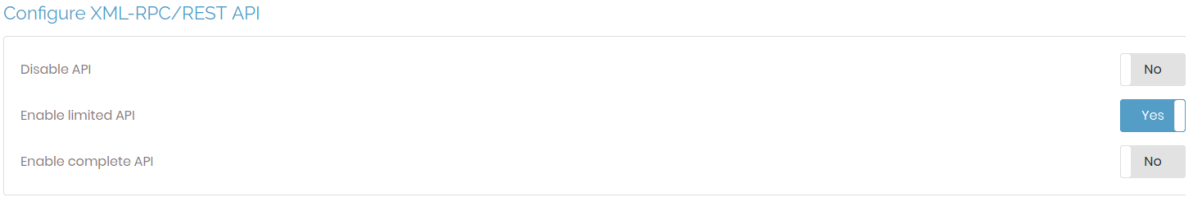
Config Data channel ciphers:



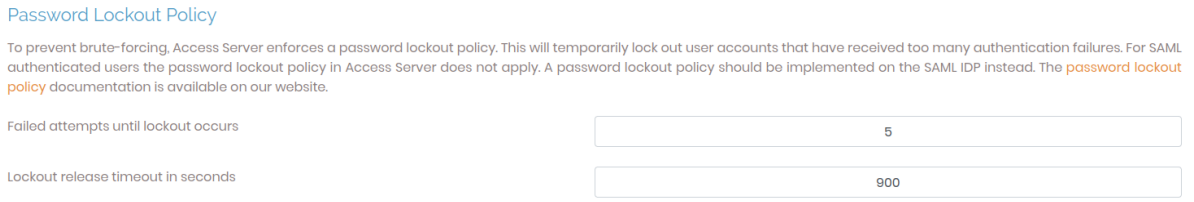
Config connection security refresh:



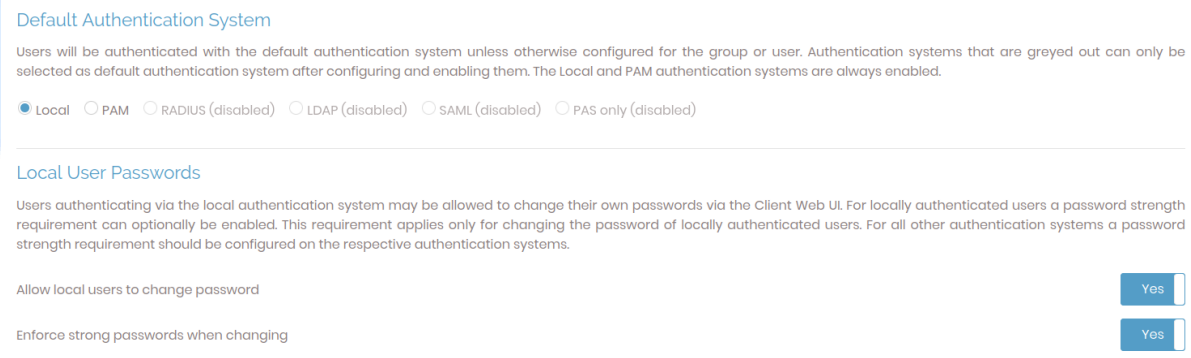
Config XML-RPC/Rest API:



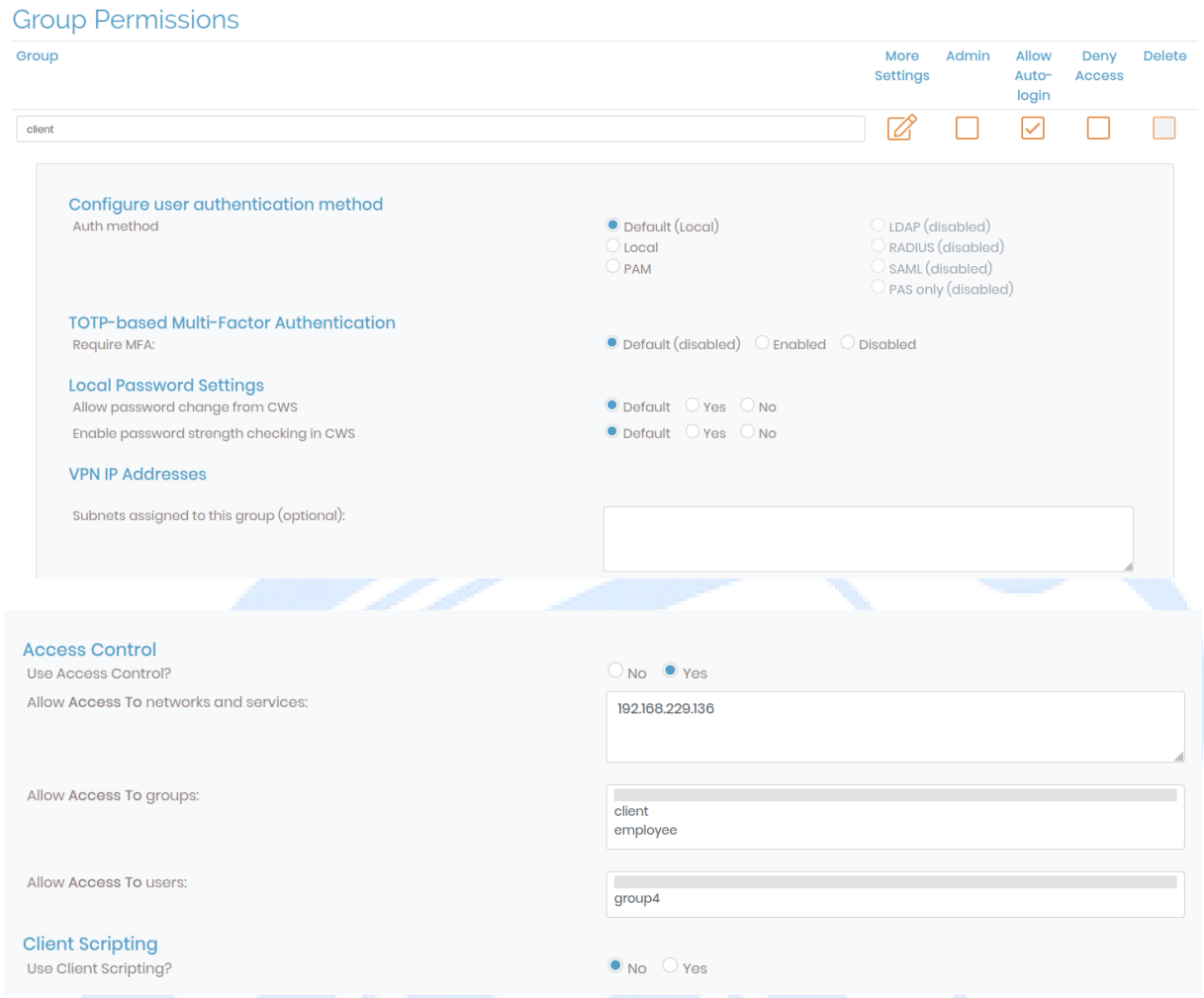
Config Password Lockout Policy



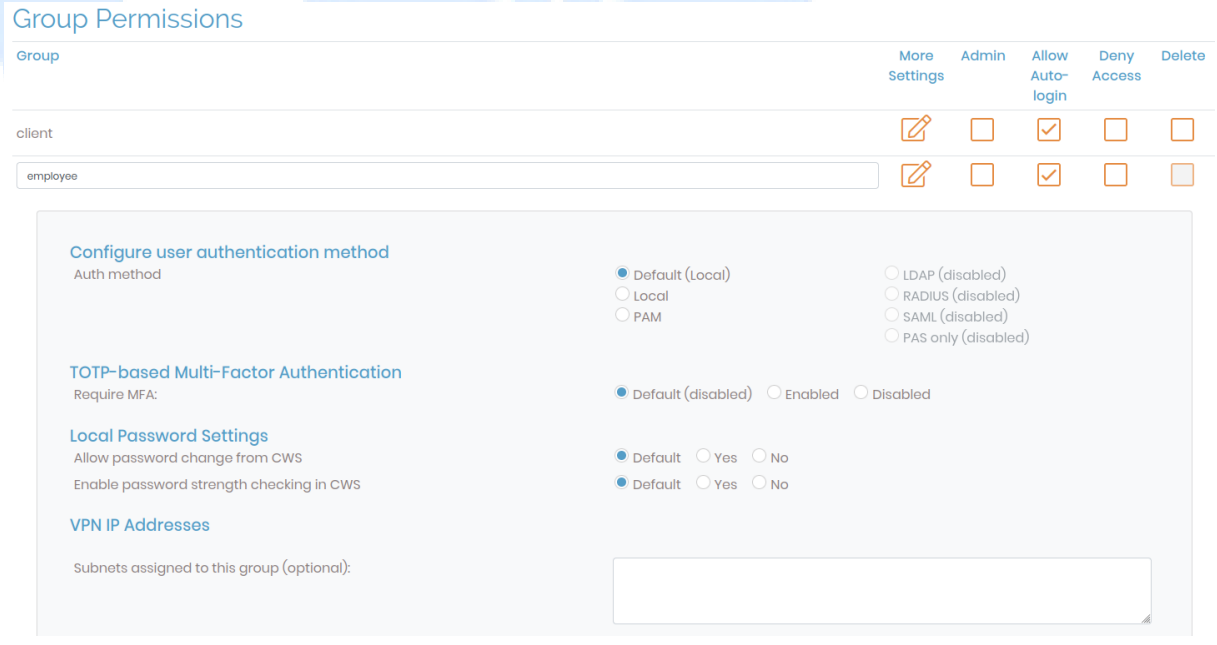
Config authentication system and user password:

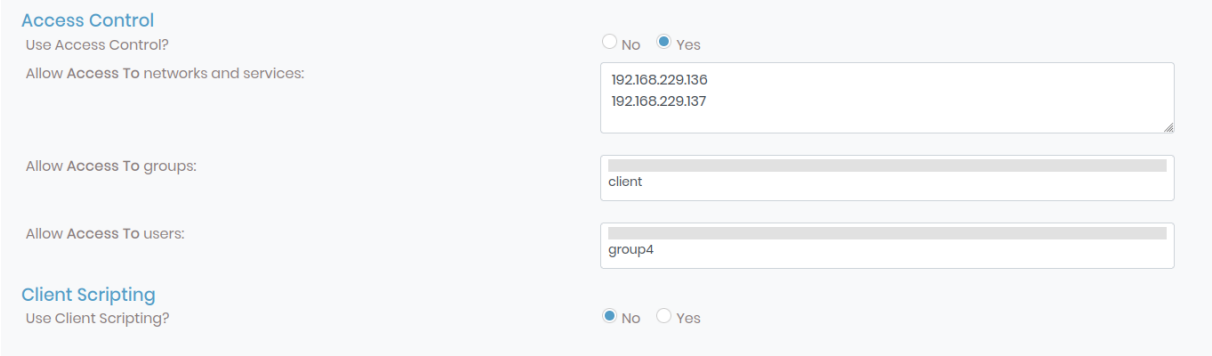


Create group “client” and allow to access to Internal Server 1 (192.168.229.136):

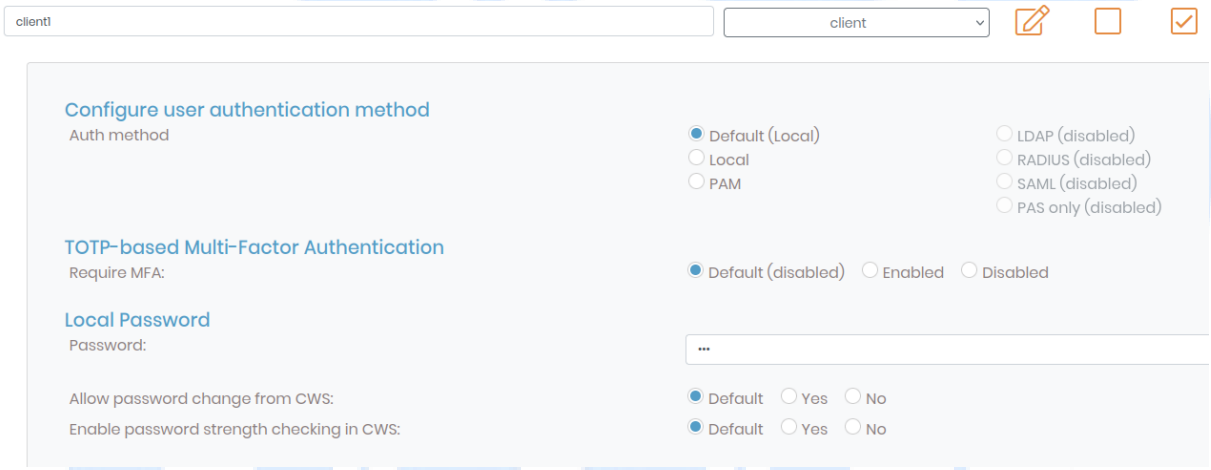


Create group “employee” and allow to access to Internal Server 1 (192.168.229.136) and Internal Server 2 (192.168.229.137):

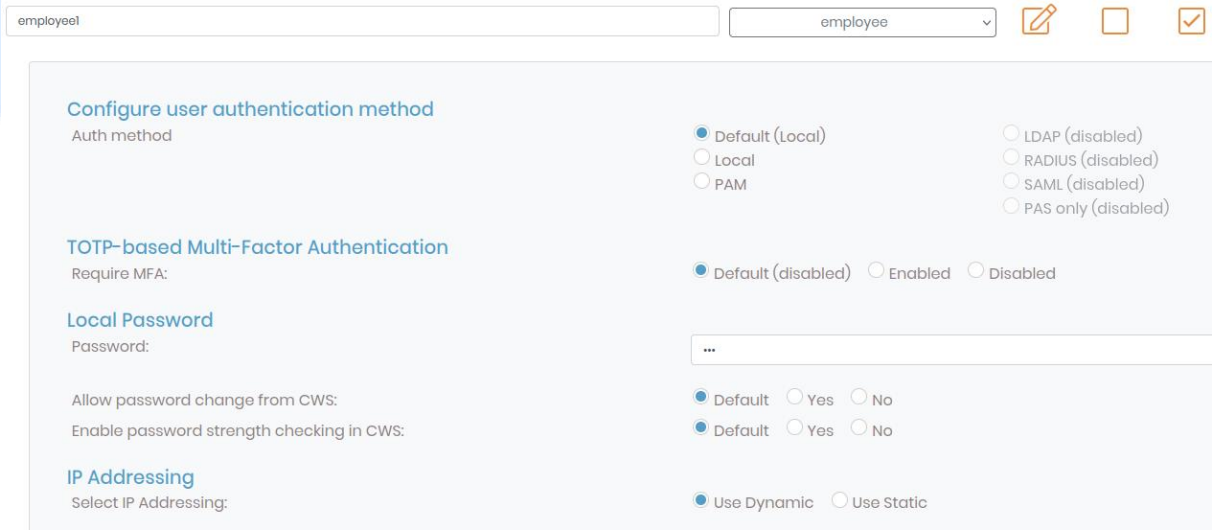




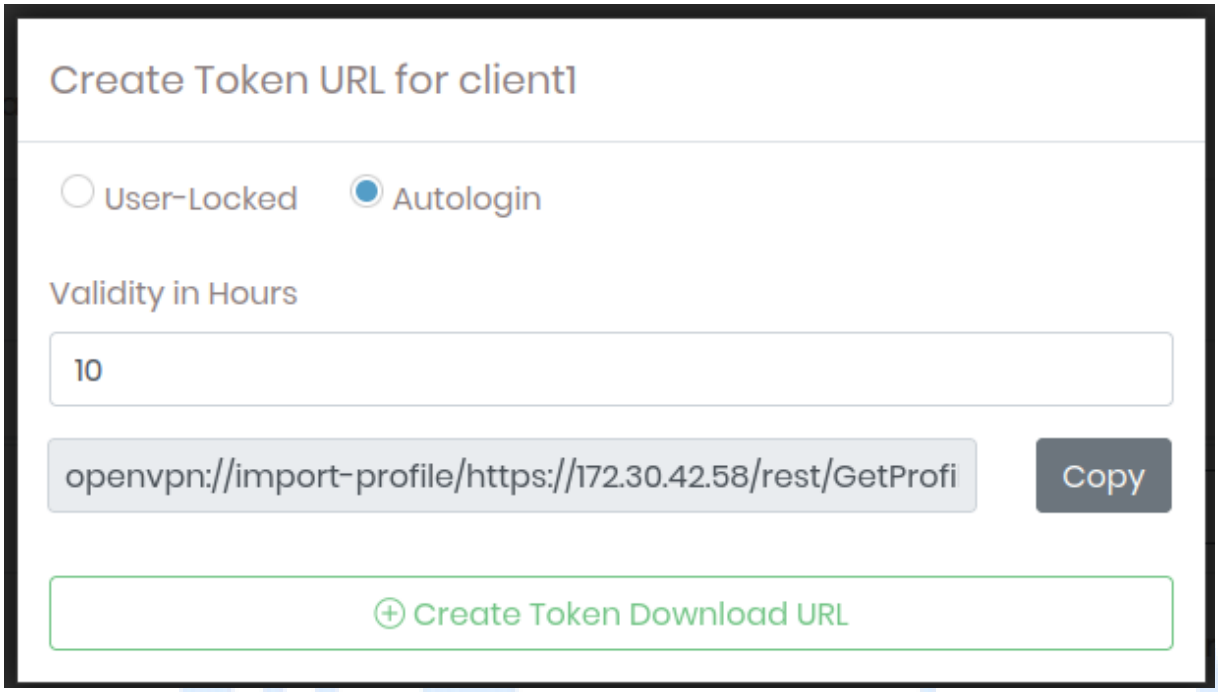
Create account “client1” and set to group “client”:

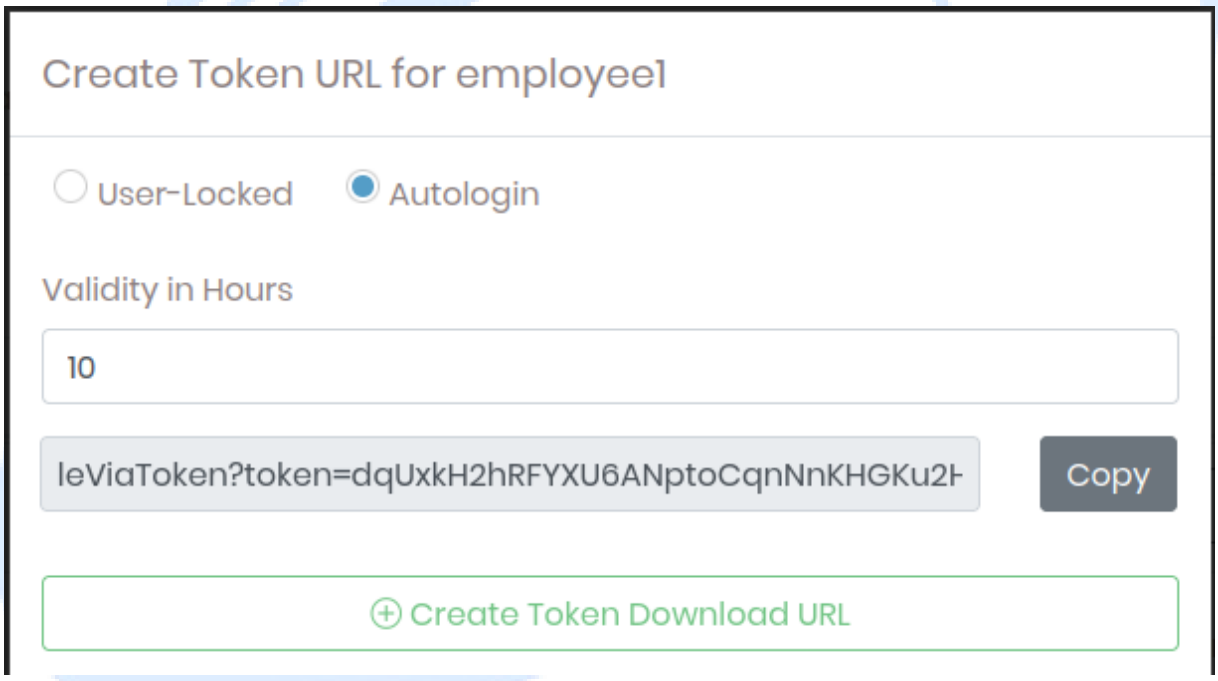


Create account “employee1” and set to group “employee”:



Create Token URL for “client1” and “employee1”:





2.3.2. Internal Server 1

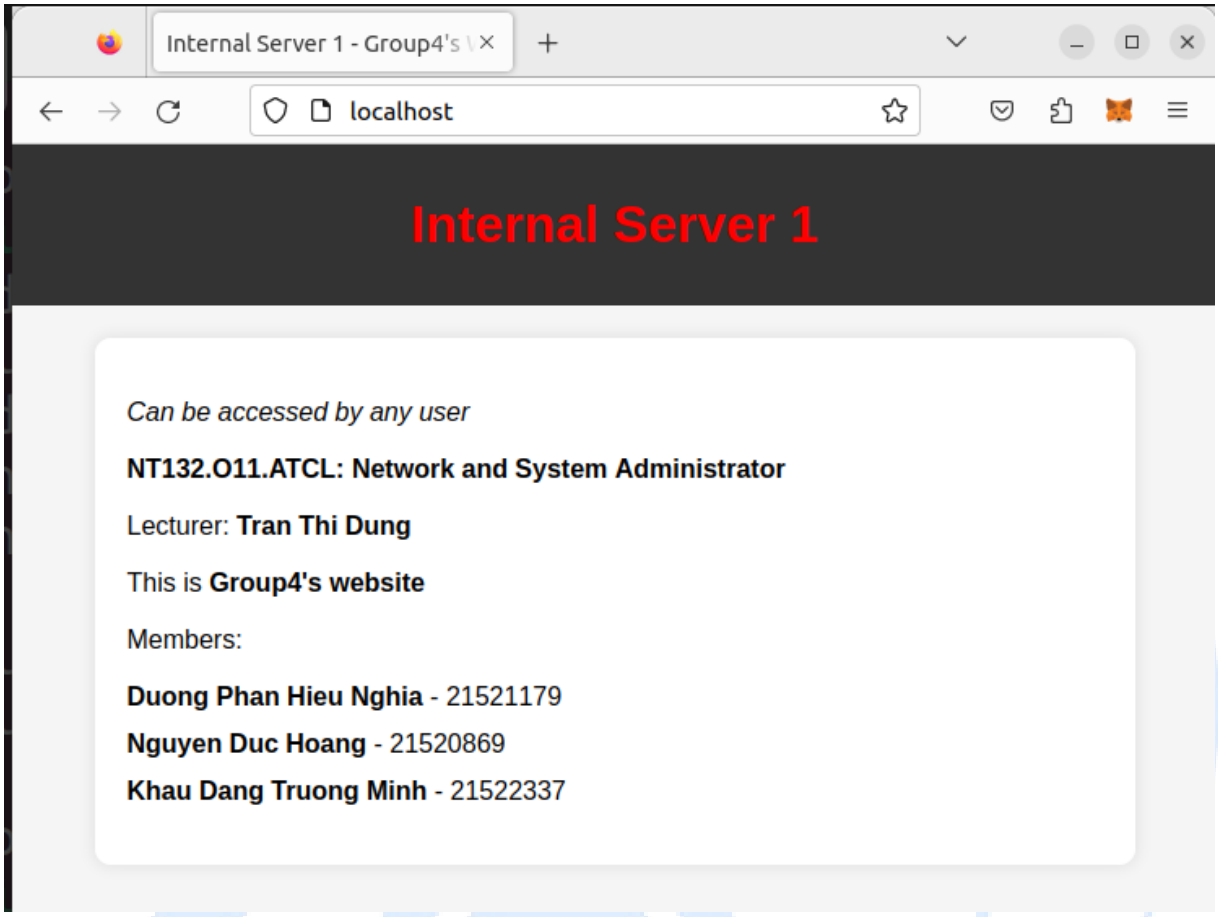
Create a simple website:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Internal Server 1 - Group4's Website</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f5f5f5;
            margin: 0;
            padding: 0;
        }
        header {
            background-color: #333;
            color: white;
            text-align: center;
            padding: 10px;
        }
        section {
            max-width: 600px;
            margin: 20px auto;
            padding: 20px;
            background-color: white;
            border-radius: 10px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
        }
        h1 {
            color: red;
        }
        em {
            font-style: italic;
        }
        strong {
            font-weight: bold;
        }
        ul {
            list-style: none;
            padding: 0;
```

```
        }
        li {
            margin-bottom: 10px;
        }
    </style>
</head>
<body>
    <header>
        <h1><strong>Internal Server 1</strong></h1>
    </header>
    <section>
        <p><em>Can be accessed by any user</em></p>
        <p><strong>NT132.O11.ATCL: Network and System
Administrator</strong></p>
        <p>Lecturer: <strong>Tran Thi Dung</strong></p>
        <p>This is <strong>Group4's website</strong></p>
        <p>Members:</p>
        <ul>
            <li><strong>Duong Phan Hieu Nghia</strong> -
21521179</li>
            <li><strong>Nguyen Duc Hoang</strong> -
21520869</li>
            <li><strong>Khau Dang Truong Minh</strong> -
21522337</li>
        </ul>
    </section>
</body>
</html>
```

View:



Create user “client”:

```
nghia@nghia-virtual-machine:~$ sudo adduser client
Adding user `client' ...
Adding new group `client' (1002) ...
Adding new user `client' (1002) with group `client' ...
The home directory `/home/client' already exists.  Not copying from `/etc/skel'.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: password updated successfully
Changing the user information for client
Enter the new value, or press ENTER for the default
        Full Name []: Client 1
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
```

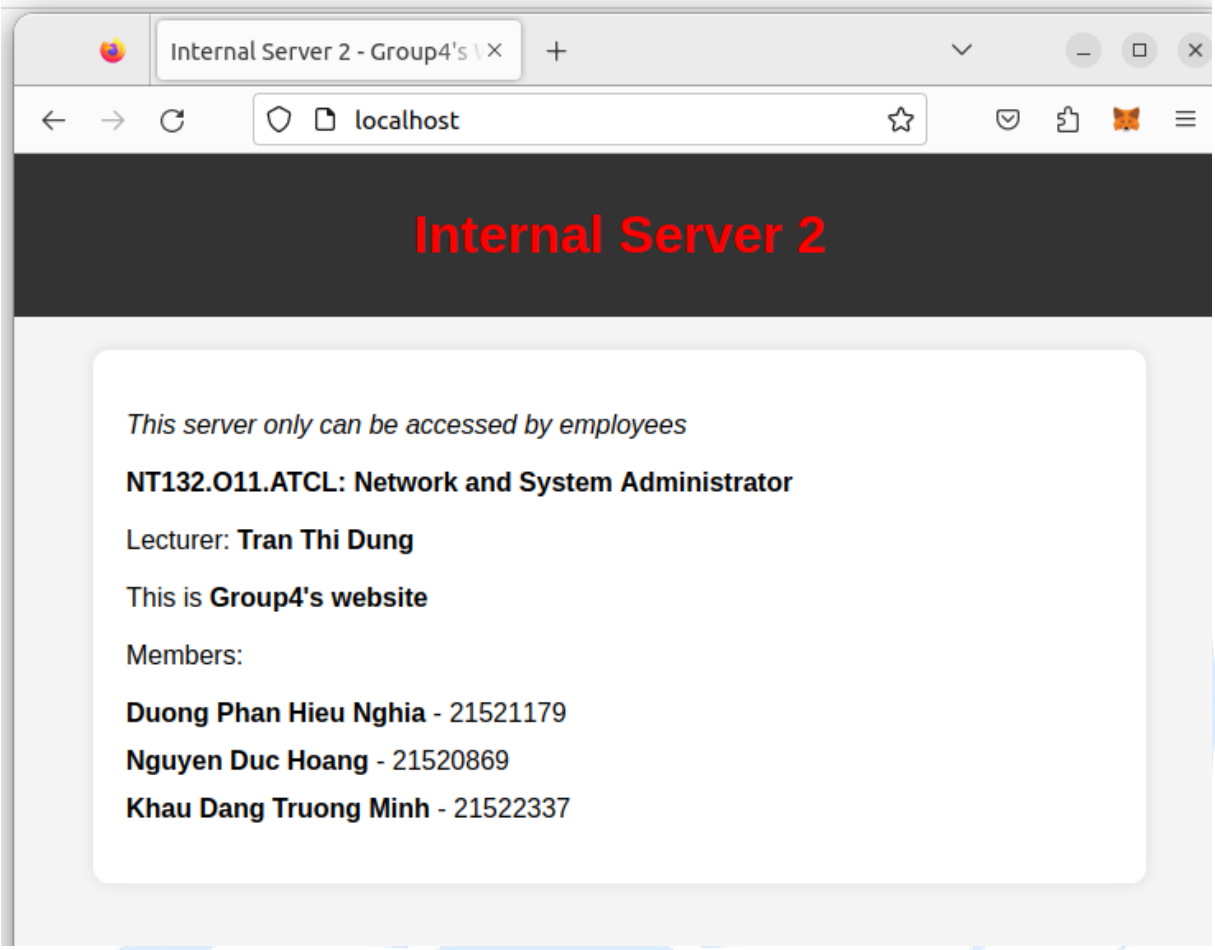
2.3.3. Internal Server 2

Create a simple website:

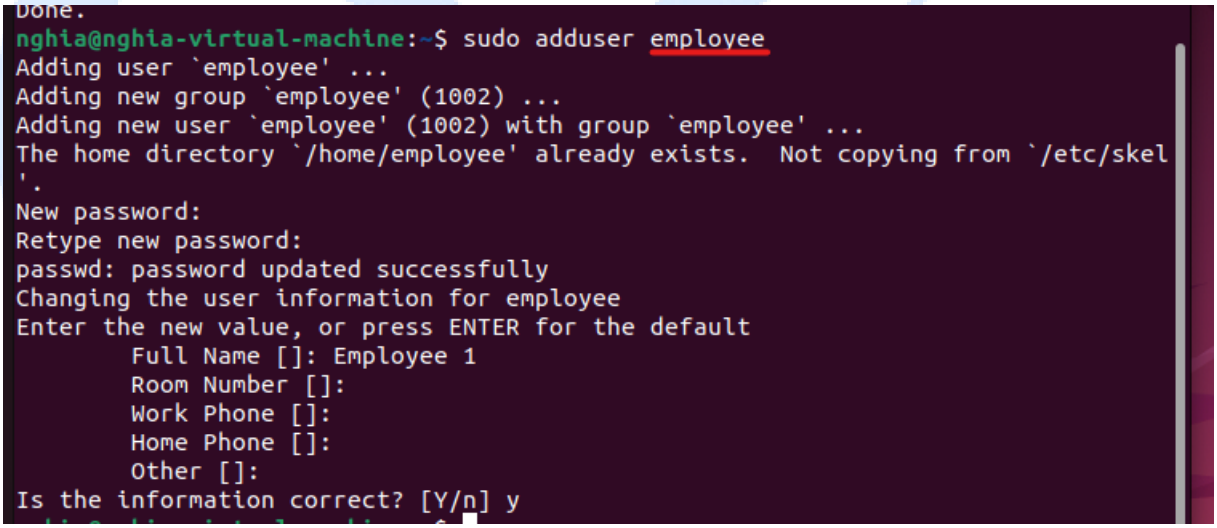
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Internal Server 2 - Group4's Website</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f5f5f5;
            margin: 0;
            padding: 0;
        }
        header {
            background-color: #333;
            color: white;
            text-align: center;
            padding: 10px;
        }
        section {
            max-width: 600px;
            margin: 20px auto;
            padding: 20px;
            background-color: white;
            border-radius: 10px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
        }
        h1 {
            color: red;
        }
        em {
            font-style: italic;
        }
        strong {
            font-weight: bold;
```

```
        }
        ul {
            list-style: none;
            padding: 0;
        }
        li {
            margin-bottom: 10px;
        }
    </style>
</head>
<body>
    <header>
        <h1><strong>Internal Server 2</strong></h1>
    </header>
    <section>
        <p><em>This server only can be accessed by
employees</em></p>
        <p><strong>NT132.O11.ATCL: Network and System
Administrator</strong></p>
        <p>Lecturer: <strong>Tran Thi Dung</strong></p>
        <p>This is <strong>Group4's website</strong></p>
        <p>Members:</p>
        <ul>
            <li><strong>Duong Phan Hieu Nghia</strong> -
21521179</li>
            <li><strong>Nguyen Duc Hoang</strong> -
21520869</li>
            <li><strong>Khau Dang Truong Minh</strong> -
21522337</li>
        </ul>
    </section>
</body>
</html>
```

View:



Create user “employee”:



2.3.4. Client

Generate key for ssh connection:

```
┌──(kali㉿kali)-[~/Desktop/OpenVPN]
└─$ ssh-keygen -t rsa -b 2048
Generating public/private rsa key pair.
Enter file in which to save the key (/home/kali/.ssh/id_rsa):
/home/kali/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/kali/.ssh/id_rsa
Your public key has been saved in /home/kali/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:eYMX65d11Blr81uqpG44rWzZkAs31HOtTG2MizdQxVE kali@kali
The key's randomart image is:
+---[RSA 2048]----+
|             oo+E|
|           . . =|
|          ...  = =o|
|         .o+o+ *.o|
|        .S.=B +. +|
|       . =+..*o oo|
|        o O..+ ...  |
|       .* ++ .      |
|       .o=o .       |
+----[SHA256]-----+
```

Copy ssh key to user “client” on Internal Server 1:

```
┌──(kali㉿kali)-[~]
└─$ ssh-copy-id client@192.168.229.136

/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/kali/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
client@192.168.229.136's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'client@192.168.229.136'"
and check to make sure that only the key(s) you wanted were added.
```

Copy ssh key to user “employee” on Internal Server 2:

```
┌──(kali㉿kali)-[~]
└─$ ssh-copy-id employee@192.168.229.137

/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/kali/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
employee@192.168.229.137's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'employee@192.168.229.137'"
and check to make sure that only the key(s) you wanted were added.
```

Download openvpn connection for client1:

```
┌──(kali㉿kali)-[~/Desktop/OpenVPN]
└─$ curl -k -o client1.ovpn "https://172.30.42.58/rest/GetProfileViaToken?token=O5yKAimJFK1fM7ZbSFg7bn8PMHt4G25H"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  6064  100  6064    0     0  61980      0 --:--:-- --:--:-- --:--:-- 62515
```

Download openvpn connection for employee1:

```
┌──(kali㉿kali)-[~/Desktop/OpenVPN]
└─$ curl -k -o employee1.ovpn "https://172.30.42.58/rest/GetProfileViaToken?token=dqUxkH2hRFYXU6ANptoCqnNnKHGKu2Hw"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  6070  100  6070    0     0    98k      0 --:--:-- --:--:-- --:--:--   98k
```

III. RESULT AND CONCLUSION

3.1. Result

3.1.1. OpenVPN Server

Check NICs:

```
nghia@nghia-virtual-machine:~$ ifconfig
as0t0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 172.27.224.1  netmask 255.255.252.0  destination 172.27.224.1
        inet6 fe80::6531:893:b9c2:6013  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 200  (UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 10  bytes 480 (480.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

as0t1: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 172.27.228.1  netmask 255.255.252.0  destination 172.27.228.1
        inet6 fe80::8d17:286b:647f:cfd0  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 200  (UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 9  bytes 432 (432.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
as0t2: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 172.27.232.1  netmask 255.255.252.0  destination 172.27.232.1
        inet6 fe80::2b9c:ab41:33f3:bf01  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 200  (UNSPEC)
        RX packets 119  bytes 27668 (27.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 108  bytes 24310 (24.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

as0t3: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 172.27.236.1  netmask 255.255.252.0  destination 172.27.236.1
        inet6 fe80::67b4:b515:66dc:5201  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 200  (UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 9  bytes 432 (432.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.30.42.58  netmask 255.255.0.0  broadcast 172.30.255.255
        inet6 fe80::1195:ebcd:7ce5:848  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:5e:3a:17  txqueuelen 1000  (Ethernet)
        RX packets 317560  bytes 88632524 (88.6 MB)
        RX errors 6  dropped 541  overruns 0  frame 0
        TX packets 5924  bytes 582101 (582.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 19  base 0x2000

ens34: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.229.132  netmask 255.255.255.0  broadcast 192.168.229.255
        inet6 fe80::a83c:e238:9561:e681  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:5e:3a:21  txqueuelen 1000  (Ethernet)
        RX packets 368  bytes 55076 (55.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 384  bytes 52655 (52.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 16  base 0x2080
```

Check IP route:

```
nghia@nghia-virtual-machine:~$ ip route
default via 172.30.0.1 dev ens33 proto dhcp metric 100
169.254.0.0/16 dev ens34 scope link metric 1000
172.27.224.0/22 dev as0t0 proto kernel scope link src 172.27.224.1
172.27.228.0/22 dev as0t1 proto kernel scope link src 172.27.228.1
172.27.232.0/22 dev as0t2 proto kernel scope link src 172.27.232.1
172.27.236.0/22 dev as0t3 proto kernel scope link src 172.27.236.1
172.30.0.0/16 dev ens33 proto kernel scope link src 172.30.42.58 metric 100
192.168.229.0/24 dev ens34 proto kernel scope link src 192.168.229.132 metric 101
```

Check routing table:

```
nghia@nghia-virtual-machine:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         172.30.0.1      0.0.0.0         UG    100    0        0 ens33
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 ens34
172.27.224.0    0.0.0.0         255.255.252.0   U     0      0        0 as0t0
172.27.228.0    0.0.0.0         255.255.252.0   U     0      0        0 as0t1
172.27.232.0    0.0.0.0         255.255.252.0   U     0      0        0 as0t2
172.27.236.0    0.0.0.0         255.255.252.0   U     0      0        0 as0t3
172.30.0.0      0.0.0.0         255.255.0.0     U     100    0        0 ens33
192.168.229.0   0.0.0.0         255.255.255.0   U     101    0        0 ens34
```

3.1.2. Check account “client1”

Test ping to Internal Server 1 (192.168.229.136) and Internal Server 2 (192.168.229.137):

```
┌──(kali㉿kali)-[~]
└─$ ping 192.168.229.136
PING 192.168.229.136 (192.168.229.136) 56(84) bytes of data.
^C
— 192.168.229.136 ping statistics —
4 packets transmitted, 0 received, 100% packet loss, time 3057ms


┌──(kali㉿kali)-[~]
└─$ ping 192.168.229.137
PING 192.168.229.137 (192.168.229.137) 56(84) bytes of data.
^C
— 192.168.229.137 ping statistics —
3 packets transmitted, 0 received, 100% packet loss, time 2028ms
```

➔ Fail.

Connect OpenVPN with client1:

```
┌──(kali㉿kali)-[~/Desktop/OpenVPN]
└─$ sudo openvpn client1.ovpn
2023-12-04 22:54:38 DEPRECATED OPTION: --cipher set to 'AES-256-CBC' but missing in --data-ciphers (AES-256-GCM
:AES-128-GCM:CHACHA20-POLY1305). OpenVPN ignores --cipher for cipher negotiations.
2023-12-04 22:54:38 Note: Kernel support for ovpn-dco missing, disabling data channel offload.
2023-12-04 22:54:38 OpenVPN 2.6.7 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO]
 [AEAD] [DCO]
2023-12-04 22:54:38 library versions: OpenSSL 3.0.11 19 Sep 2023, LZO 2.10
2023-12-04 22:54:38 DCO version: N/A
2023-12-04 22:54:38 TCP/UDP: Preserving recently used remote address: [AF_INET]172.30.42.58:1194
2023-12-04 22:54:38 Socket Buffers: R=[212992→212992] S=[212992→212992]
2023-12-04 22:54:38 UDPv4 link local: (not bound)
2023-12-04 22:54:38 UDPv4 link remote: [AF_INET]172.30.42.58:1194
2023-12-04 22:54:38 TLS: Initial packet from [AF_INET]172.30.42.58:1194, sid=8a9dd4ec ccf200b8
2023-12-04 22:54:38 net_route_v4_best_gw query: dst 0.0.0.0
2023-12-04 22:54:38 net_route_v4_best_gw result: via 172.30.0.1 dev eth0
2023-12-04 22:54:38 VERIFY OK: depth=1, CN=OpenVPN CA
2023-12-04 22:54:38 VERIFY KU OK
```

Check NICs:

```
tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 172.27.240.3  netmask 255.255.248.0  destination 172.27.240.3
        inet6 fe80::11df:860e:7f55:8108  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500  (UNSPEC)
        RX packets 4  bytes 384 (384.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4  bytes 192 (192.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Check IP route:

```
┌──(kali㉿kali)-[~]
└─$ ip route
0.0.0.0/1 via 172.27.240.1 dev tun0
default via 172.30.0.1 dev eth0 proto dhcp src 172.30.70.250 metric 100
128.0.0.0/1 via 172.27.240.1 dev tun0
172.27.240.0/21 dev tun0 proto kernel scope link src 172.27.240.3
172.30.0.0/16 dev eth0 proto kernel scope link src 172.30.70.250 metric 100
```

Check routing table:

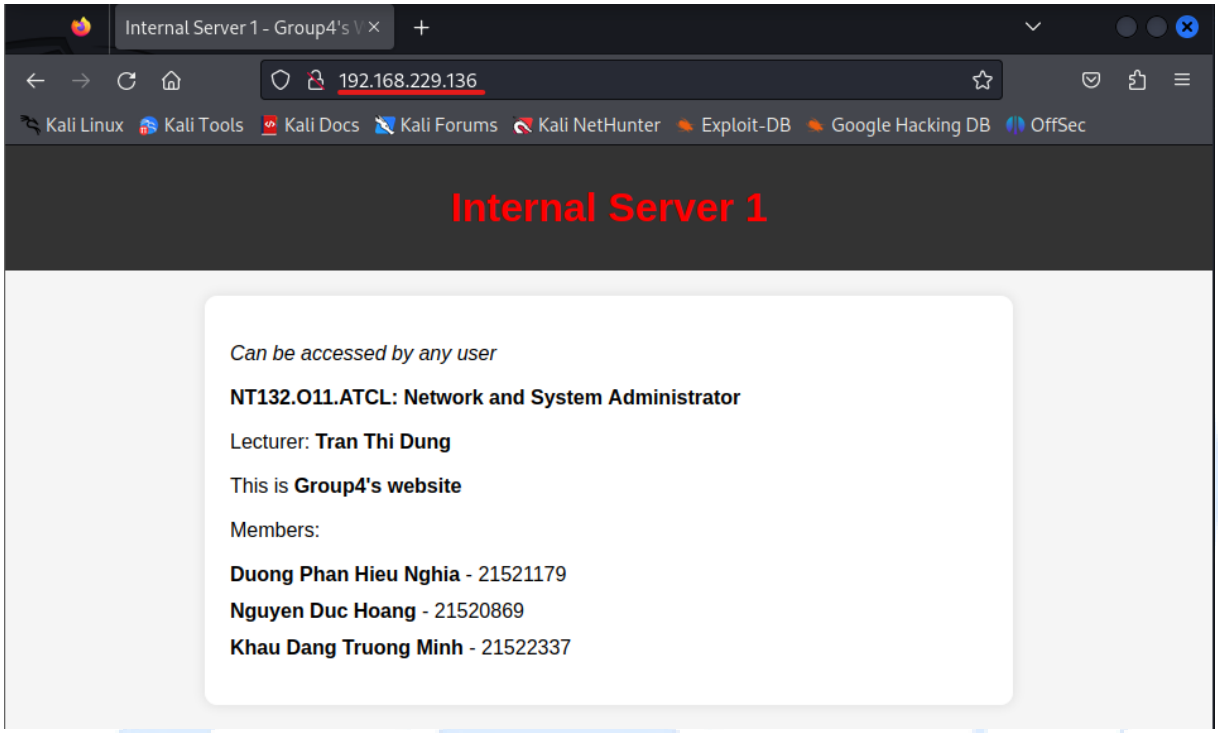
```
┌──(kali㉿kali)-[~]
└─$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         172.27.240.1    128.0.0.0       UG    0      0        0 tun0
0.0.0.0         172.30.0.1      0.0.0.0         UG    100    0        0 eth0
128.0.0.0       172.27.240.1    128.0.0.0       UG    0      0        0 tun0
172.27.240.0    0.0.0.0         255.255.248.0   U     0      0        0 tun0
172.30.0.0      0.0.0.0         255.255.0.0     U     100    0        0 eth0
```

Ping to Internal Server 1 (192.168.229.136):

```
┌──(kali㉿kali)-[~]
└─$ ping 192.168.229.136
PING 192.168.229.136 (192.168.229.136) 56(84) bytes of data.
64 bytes from 192.168.229.136: icmp_seq=1 ttl=63 time=5.77 ms
64 bytes from 192.168.229.136: icmp_seq=2 ttl=63 time=1.63 ms
64 bytes from 192.168.229.136: icmp_seq=3 ttl=63 time=2.32 ms
^C
── 192.168.229.136 ping statistics ──
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 1.634/3.239/5.766/1.808 ms
```

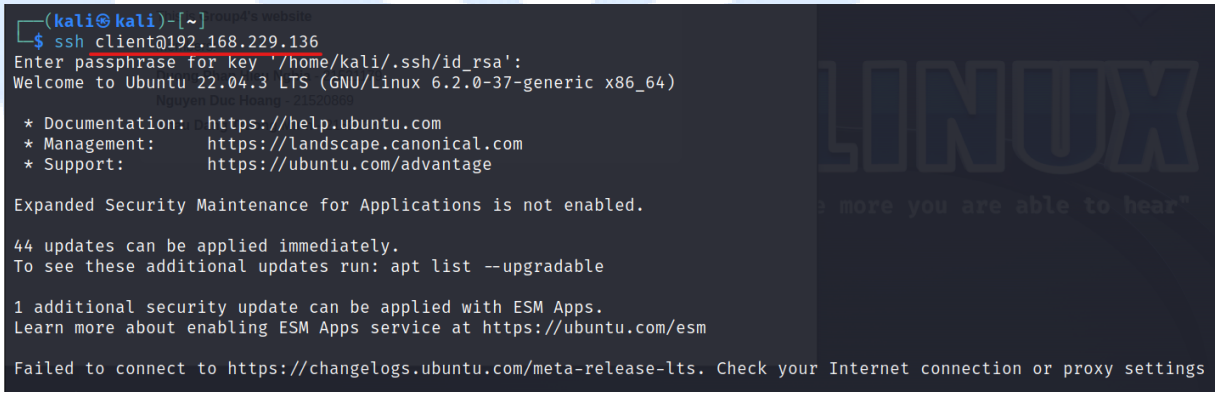
➔ Successful.

Access to Website on Internal Server 1 (192.168.229.136):



➔ Successful.

Connect ssh to Internal Server 1 (192.168.229.136) with key:



```
client@nghia-virtual-machine:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.229.136  netmask 255.255.255.0  broadcast 192.168.229.255
        inet6 fe80::be11:9492:81e6:d80f  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:47:1c:73  txqueuelen 1000  (Ethernet)
        RX packets 538  bytes 72578 (72.5 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 292  bytes 48168 (48.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 19  base 0x2000

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 63627  bytes 4718417 (4.7 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 63627  bytes 4718417 (4.7 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

➔ Successful.

Ping to Internal Server 2 (192.168.229.137):

```
┌──(kali㉿kali)-[~]
└─$ ping 192.168.229.137
PING 192.168.229.137 (192.168.229.137) 56(84) bytes of data.
^C
── 192.168.229.137 ping statistics ──
3 packets transmitted, 0 received, 100% packet loss, time 2045ms
```

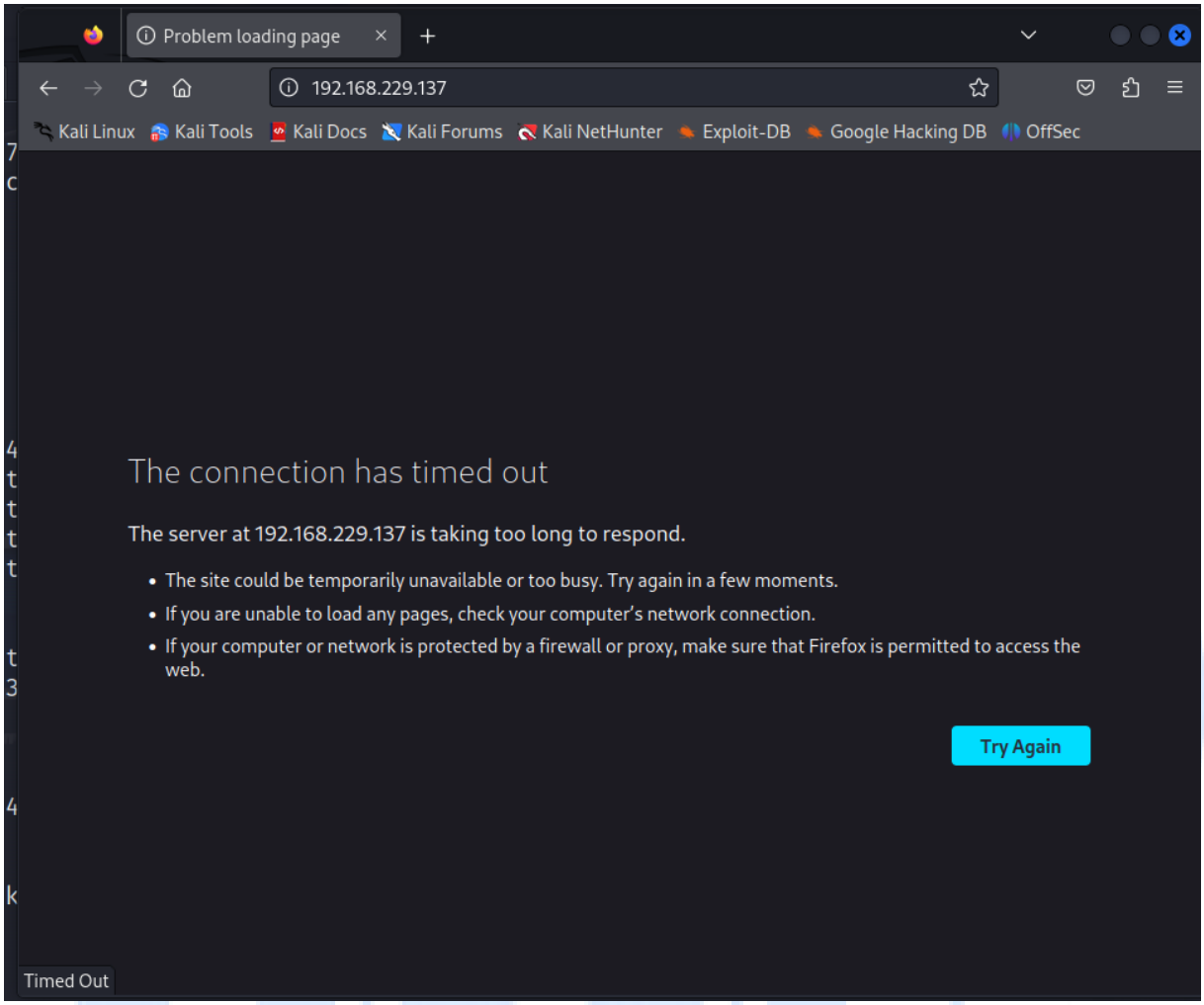
➔ Fail. → Access control.

Try to connect ssh to Internal Server 2 (192.168.229.137):

```
┌──(kali㉿kali)-[~]
└─$ ssh employee@192.168.229.137
```

➔ Can not make a connection. → Access control.

Try to access website on Internal Server 2 (192.168.229.137:



➔ Can not load the website. → Access control.

3.1.3. Check account "employee1"

Connect OpenVPN with employee1:

```
┌──(kali㉿kali)-[~/Desktop/OpenVPN]
└─$ sudo openvpn employee1.ovpn
2023-12-04 23:03:26 DEPRECATED OPTION: --cipher set to 'AES-256-CBC' but missing in --data-ciphers (AES-256-GCM
:AES-128-GCM:CHACHA20-POLY1305). OpenVPN ignores --cipher for cipher negotiations.
2023-12-04 23:03:26 Note: Kernel support for ovpn-dco missing, disabling data channel offload.
2023-12-04 23:03:26 OpenVPN 2.6.7 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO]
 [AEAD] [DCO]
2023-12-04 23:03:26 library versions: OpenSSL 3.0.11 19 Sep 2023, LZO 2.10
2023-12-04 23:03:26 DCO version: N/A
2023-12-04 23:03:26 TCP/UDP: Preserving recently used remote address: [AF_INET]172.30.42.58:1194
2023-12-04 23:03:26 Socket Buffers: R=[212992→212992] S=[212992→212992]
2023-12-04 23:03:26 UDPv4 link local: (not bound)
2023-12-04 23:03:26 UDPv4 link remote: [AF_INET]172.30.42.58:1194
2023-12-04 23:03:26 TLS: Initial packet from [AF_INET]172.30.42.58:1194, sid=c65636ff 37f0d3a2
2023-12-04 23:03:26 net_route_v4_best_gw query: dst 0.0.0.0
2023-12-04 23:03:26 net_route_v4_best_gw result: via 172.30.0.1 dev eth0
2023-12-04 23:03:26 VERIFY OK: depth=1, CN=OpenVPN CA
2023-12-04 23:03:26 VERIFY KU OK
```

Check NICs:

```
tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 172.27.248.3  netmask 255.255.248.0  destination 172.27.248.3
        inet6 fe80::ab29:f461:9170:4c85  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500  (UNSPEC)
        RX packets 48  bytes 11395 (11.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 100  bytes 13654 (13.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Check IP route:

```
┌──(kali㉿kali)-[~]
└─$ ip route
0.0.0.0/1 via 172.27.248.1 dev tun0
default via 172.30.0.1 dev eth0 proto dhcp src 172.30.70.250 metric 100
128.0.0.0/1 via 172.27.248.1 dev tun0
172.27.248.0/21 dev tun0 proto kernel scope link src 172.27.248.3
172.30.0.0/16 dev eth0 proto kernel scope link src 172.30.70.250 metric 100
```

Check routing table:

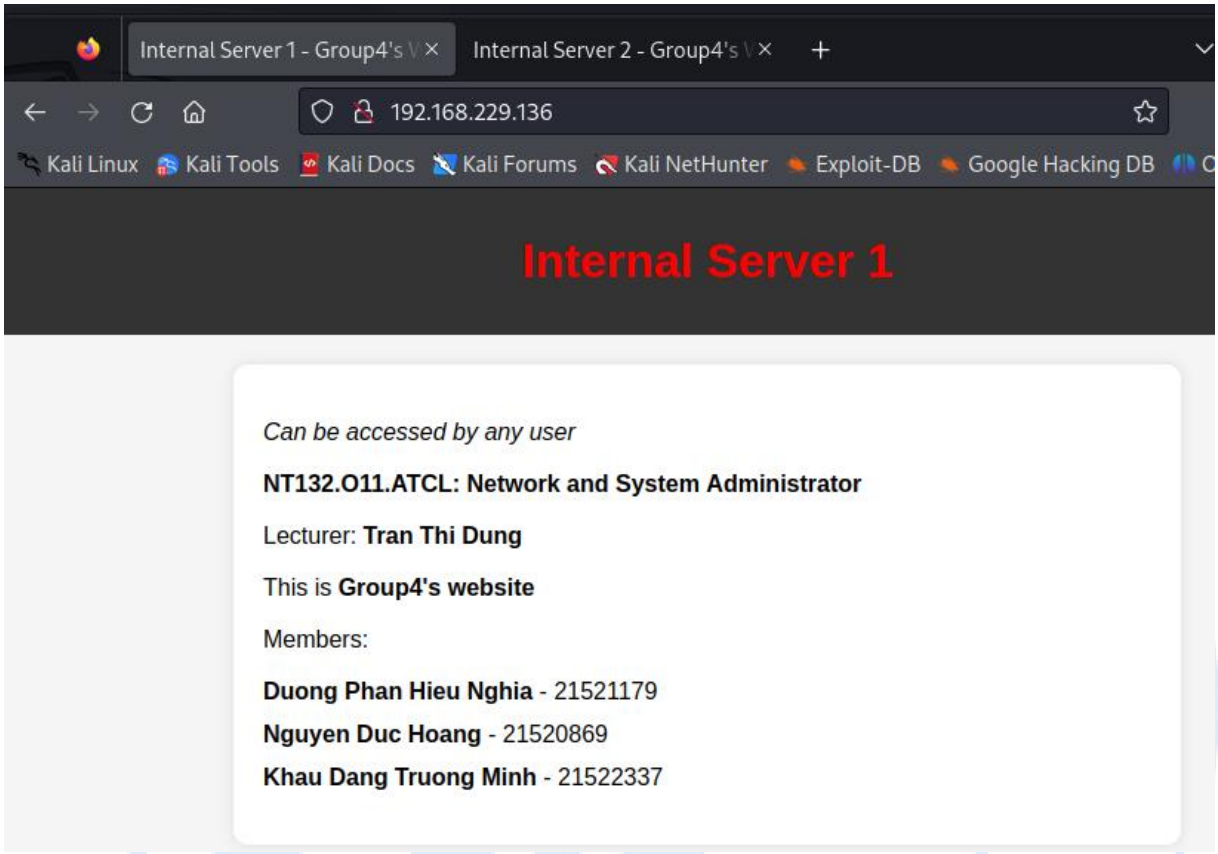
```
┌──(kali㉿kali)-[~]
└─$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         172.27.248.1    128.0.0.0       UG    0      0        0 tun0
0.0.0.0         172.30.0.1      0.0.0.0         UG    100    0        0 eth0
128.0.0.0       172.27.248.1    128.0.0.0       UG    0      0        0 tun0
172.27.248.0    0.0.0.0         255.255.248.0   U     0      0        0 tun0
172.30.0.0      0.0.0.0         255.255.0.0     U     100    0        0 eth0
```

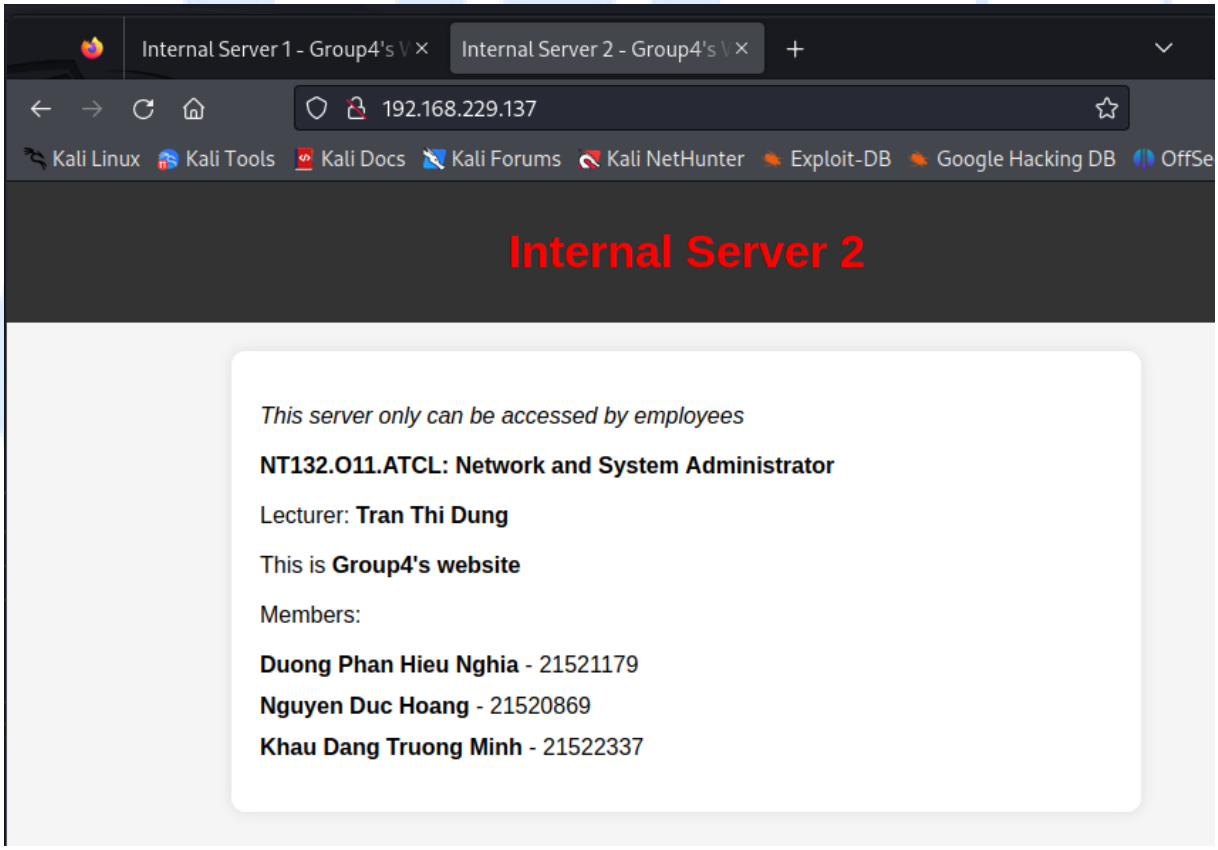
Ping to Internal Server 1 (192.168.229.136) and Internal Server 2 (192.168.229.137):

```
┌──(kali㉿kali)-[~]
└─$ ping 192.168.229.136
PING 192.168.229.136 (192.168.229.136) 56(84) bytes of data.
64 bytes from 192.168.229.136: icmp_seq=1 ttl=63 time=2.33 ms
64 bytes from 192.168.229.136: icmp_seq=2 ttl=63 time=2.62 ms
64 bytes from 192.168.229.136: icmp_seq=3 ttl=63 time=2.89 ms
^C
── 192.168.229.136 ping statistics ──
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 2.333/2.615/2.893/0.228 ms

┌──(kali㉿kali)-[~]
└─$ ping 192.168.229.137
PING 192.168.229.137 (192.168.229.137) 56(84) bytes of data.
64 bytes from 192.168.229.137: icmp_seq=1 ttl=63 time=7.92 ms
64 bytes from 192.168.229.137: icmp_seq=2 ttl=63 time=2.35 ms
64 bytes from 192.168.229.137: icmp_seq=3 ttl=63 time=3.02 ms
^C
── 192.168.229.137 ping statistics ──
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 2.353/4.429/7.917/2.481 ms
```

➔ Successful both servers.

Access to website on Internal Server 1 (192.168.229.136) and Internal Server 2 (192.168.229.137):





➔ Success both servers.

Connect ssh to Internal Server 1 (192.168.229.136):

```
┌──(kali㉿kali)-[~]
└─$ ssh client@192.168.229.136
Enter passphrase for key '/home/kali/.ssh/id_rsa':
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-37-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

Expanded Security Maintenance for Applications is not enabled.

44 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

1 additional security update can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Tue Dec  5 10:57:58 2023 from 192.168.229.132
```

```
client@nghia-virtual-machine:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.229.136  netmask 255.255.255.0  broadcast 192.168.229.255
        inet6 fe80::be11:9492:81e6:d80f  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:47:1c:73  txqueuelen 1000  (Ethernet)
        RX packets 805  bytes 103062 (103.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 359  bytes 61267 (61.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 19  base 0x2000

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 85132  bytes 6306243 (6.3 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 85132  bytes 6306243 (6.3 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

➔ Successful.

Connect ssh to Internal Server 2 (192.168.229.137):

```
┌──(kali㉿kali)-[~]
└─$ ssh employee@192.168.229.137
Enter passphrase for key '/home/kali/.ssh/id_rsa':
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-37-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

Expanded Security Maintenance for Applications is not enabled.

45 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

1 additional security update can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

employee@nghia-virtual-machine:~$
```

```
employee@nghia-virtual-machine:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.229.137  netmask 255.255.255.0  broadcast 192.168.229.255
        inet6 fe80::20c:29ff:fe03:8bde  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:03:8b:de  txqueuelen 1000  (Ethernet)
        RX packets 792  bytes 90839 (90.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4684  bytes 401095 (401.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 19  base 0x2000

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 4578  bytes 375074 (375.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4578  bytes 375074 (375.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

➔ Successful.

3.1.4. Server Log

Node	Username	Start Time	Duration	Service	Real IP	VPN IP	Proto	Port	Bytes In	Bytes Out	Error	Version	Gui Version	Platform
nghia-virtual-machine	employee1	11/26/23 11:53	0d 00:00	VPN	192.168.0.108	172.27.248.3	UDP	1194	12.77 KB	10.80 KB		2.6.3		linux
nghia-virtual-machine	client1	11/26/23 11:55	0d 00:01	VPN	192.168.0.108	172.27.240.10	UDP	1194	30.12 KB	19.40 KB		2.6.3		linux
nghia-virtual-machine	employee1	11/26/23 11:58	0d 00:02	VPN	192.168.0.108	172.27.248.4	UDP	1194	24.41 KB	32.12 KB		2.6.3		linux
nghia-virtual-machine	openvpn	11/26/23 12:01		WEB_ADMIN	192.168.0.182						local auth failed: password verification failed			
nghia-virtual-machine	openvpn	11/26/23 12:01		WEB_ADMIN	192.168.0.182									
nghia-virtual-machine	openvpn	11/26/23 12:02		WEB_ADMIN	192.168.0.182									
nghia-virtual-machine	employee1	11/26/23 12:02		XML_API	192.168.0.108									
nghia-virtual-machine	employee1	11/26/23 12:03	0d 00:01	VPN	192.168.0.108	172.27.248.5	UDP	1194	35.81 KB	32.28 KB		2.6.3		linux
nghia-virtual-machine	openvpn	11/26/23 12:53		WEB_ADMIN	192.168.0.189									
nghia-virtual-machine	openvpn	11/26/23 12:59		WEB_ADMIN	192.168.0.144									
nghia-virtual-machine	openvpn	11/26/23 13:17		WEB_ADMIN	192.168.0.144									
nghia-virtual-machine	employee1	11/26/23 13:18		XML_API	192.168.0.108									
nghia-virtual-machine	employee1	11/26/23 13:18	0d 00:05	VPN	192.168.0.108	172.27.248.6	UDP	1194	129.96 KB	1.12 MB		2.6.3		linux
nghia-virtual-machine	openvpn	12/04/23 08:45		WEB_ADMIN	10.0.2.128									
nghia-virtual-machine	employee1	12/04/23 08:48		XML_API	10.0.2.135									
nghia-virtual-machine	client1	12/04/23 08:49		XML_API	10.0.2.135									
nghia-virtual-machine	group4	12/05/23 10:15		WEB_ADMIN	172.30.42.58						local auth failed: password verification failed			
nghia-virtual-machine	group4	12/05/23 10:15		WEB_ADMIN	172.30.42.58									
nghia-virtual-machine	client1	12/05/23 10:39		XML_API	172.30.70.250									
nghia-virtual-machine	employee1	12/05/23 10:40		XML_API	172.30.70.250									
nghia-virtual-machine	client1	12/05/23 10:49	0d 00:01	VPN	172.30.70.250	172.27.240.2	UDP	1194	17.48 KB	15.66 KB		2.6.7		linux
nghia-virtual-machine	employee1	12/05/23 10:51	0d 00:00	VPN	172.30.70.250	172.27.248.2	UDP	1194	17.35 KB	15.58 KB		2.6.7		linux
nghia-virtual-machine	client1	12/05/23 10:54	0d 00:07	VPN	172.30.70.250	172.27.240.3	UDP	1194	83.10 KB	373.73 KB		2.6.7		linux
nghia-virtual-machine	employee1	12/05/23 11:03	0d 00:05	VPN	172.30.70.250	172.27.248.3	UDP	1194	43.36 KB	41.40 KB		2.6.7		linux

3.2. Conclusion

3.2.1. References:

https://as-portal.openvpn.com/instructions/ubuntu/installation

3.2.2. Task Assignment:

Task	Nghĩa	Hoàng	Minh	Progress
General Information		✓		100%
Components			✓	100%
Operation	✓			100%
Topology			✓	100%
Installation	✓	✓	✓	100%
Configuration	✓	✓		100%
Access Control	✓	✓		100%
Slideshow	✓	✓	✓	100%
Report	✓	✓	✓	100%

3.2.3 Self-Assignment

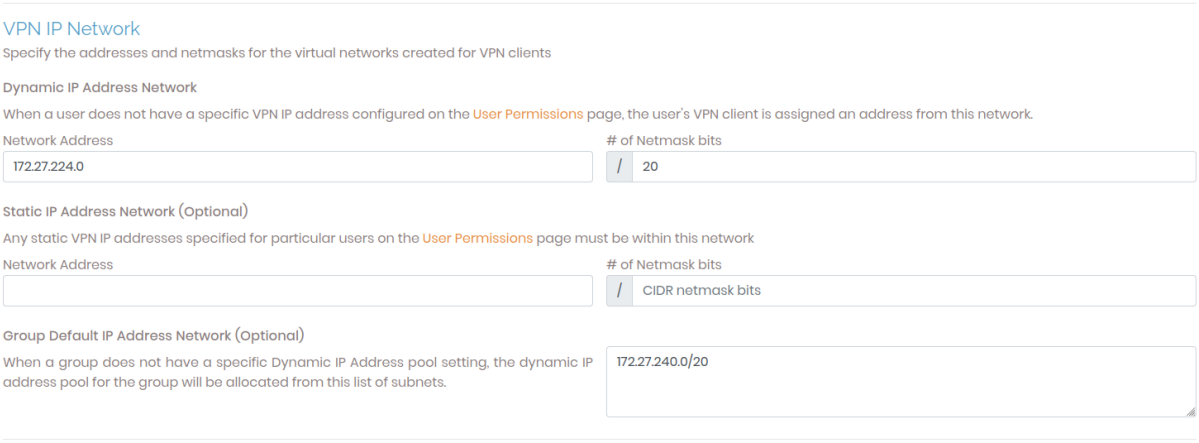
	4	3	2	1
Presentation	✓			
Theory	✓			
Demo	✓			
Report	✓			

3.2.4 Q&A

- Question 1: Can we modify the IP pool that OpenVPN provide for client connecting to the OpenVPN? If can what are the steps?

We can modify the IP pool that OpenVPN provide for client connecting to the OpenVPN.

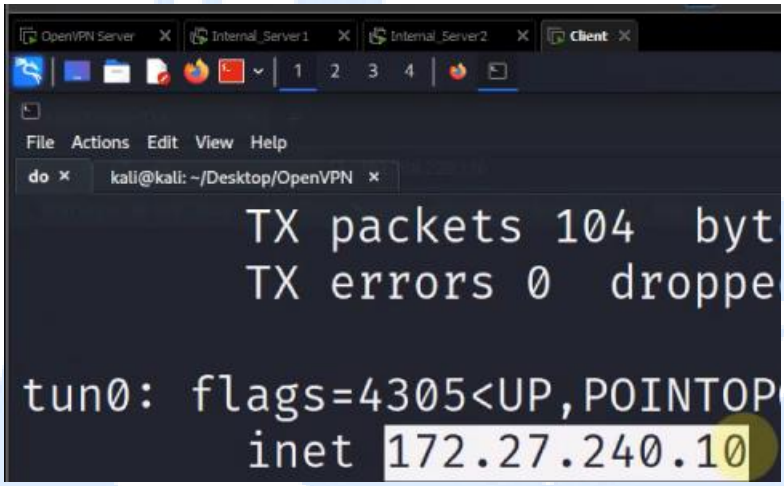
Access to OpenVPN web UI.



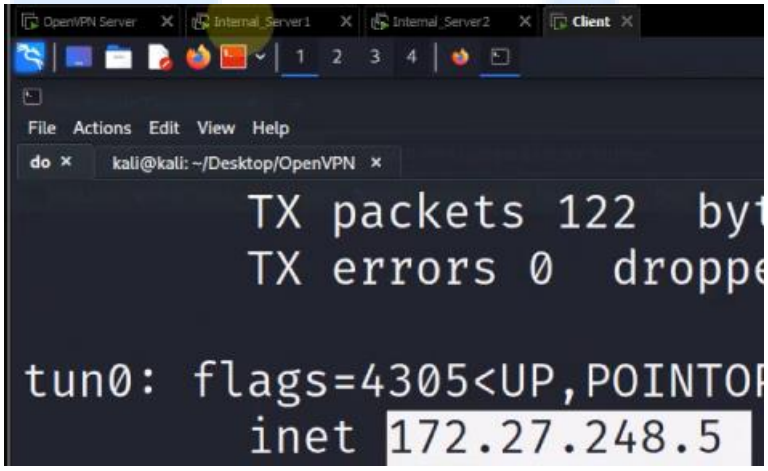
You can change the network address and netmask bits to have more or less subnet.

- Question 2: Sau khi dữ liệu routing qua OpenVPN thì có xuất hiện 2 IP 2 IP đó là gì?

Client's IP:



Employee's IP:



- Question3: How to ensure password and key security during transmission?

OpenVPN utilizes various security measures to ensure the confidentiality and integrity of passwords and keys during transmission:

Encryption: OpenVPN employs robust encryption algorithms (like AES) to scramble data, including passwords and keys, making them unreadable to anyone without the decryption key.

TLS (Transport Layer Security): It employs TLS to establish an encrypted connection, ensuring that data, including passwords and keys, transmitted between the client and server is encrypted and secure.

Authentication: OpenVPN supports different authentication methods, such as certificates, pre-shared keys, or username/password authentication. These methods ensure that both parties (client and server) can authenticate each other, preventing unauthorized access.

Diffie-Hellman Key Exchange: OpenVPN uses the Diffie-Hellman key exchange protocol to securely generate a shared secret key between the client and server. This allows them to establish a secure communication channel without explicitly transmitting the secret key.

Key Management: OpenVPN employs proper key management practices, including regular key rotation and using strong, randomly generated keys, to enhance security and minimize the risk of key compromise.

Secure Transmission Protocol: OpenVPN generally operates over UDP or TCP, providing reliable and secure data transmission, which ensures that the encrypted data, including passwords and keys, reaches its destination without being compromised.

- Question4: việc tạo giấy chứng nhận để sử dụng nên được thực hiện bởi bên khác hay nên tự tạo

We suggest using self-create certificates to gain several significant security advantages:

Stronger Security: SSH certificates provide a stronger authentication method compared to password-based logins. They use public-key encryption mechanisms to verify identity, helping to prevent various intrusion attempts like brute-force attacks or phishing.

Easier Key Management: Utilizing SSH certificates allows for more efficient key management. You can create, manage, and revoke certificates more flexibly than managing SSH passwords.

Automation Capabilities: SSH certificates support automated authentication processes, making deployment and system management more straightforward. This is especially useful for performing batch system tasks.

Better Permission Control: You can control access permissions more precisely with SSH certificates. Certificates can be configured to allow access to specific resources and specific time frames.

Two-Factor Authentication (2FA): SSH certificates can be combined with additional authentication forms like OTP (One-Time Password) or PIN codes to enhance security.