# Assignment 1 Report
# ACME-6

Francesco Aquino (1851954),
Michele Kryston (1844733),
Ralph Angelo Tancio Almoneda (1837040),
Simone Di Valerio (1835412)

Practical Network Defense, Cybersecurity Master Degree
Sapienza University of Rome
Second Semester 2022/2023

## Contents

# 1 Initial brainstorming

For the assignment we had to configure two new services on the ACME network: a VPN for the road warriors employee and a VPN tunnel between the main and the internal routers.

Before making any changes, we studied how Proxmox and OPNsense work, this took us quite some time as it was the first time we interacted with such programs. Using the credentials we received from the professor by email, we logged into Proxmox. Then we connected to the hosts "Client ext1" and ".100 PC" to open the ports in the firewalls so that we could directly connect to them. To perform the subsequent specified tasks, we worked on the two routers using OPNsense.

To make the interaction with the network nodes more easy, we used the SPICE protocol as suggested by the professor with the VirtViewer application[1].

## 1.1 What to do

We followed this path:

**1** General review of VPN and IPsec topics, study of Proxmox and OPNsense

**2** Implementation of VPN between the two routers (IPsec) to make the internal transfer of packets secure

**3** Implementation of the VPN to the main router for the Road-Warriors, so that users can connect to the local network

**4** IPv6 implementation

**5** Audit of the implementations through various tests

## 1.2 How to do it

In terms of how to do the task, we followed several guides both from the OPNsense documentation and from YouTube, and after multiple attempts we managed to get everything working. In the following paragraphs, the implementations will be explained in detail.

# 2 VPN setup for the Road-Warriors

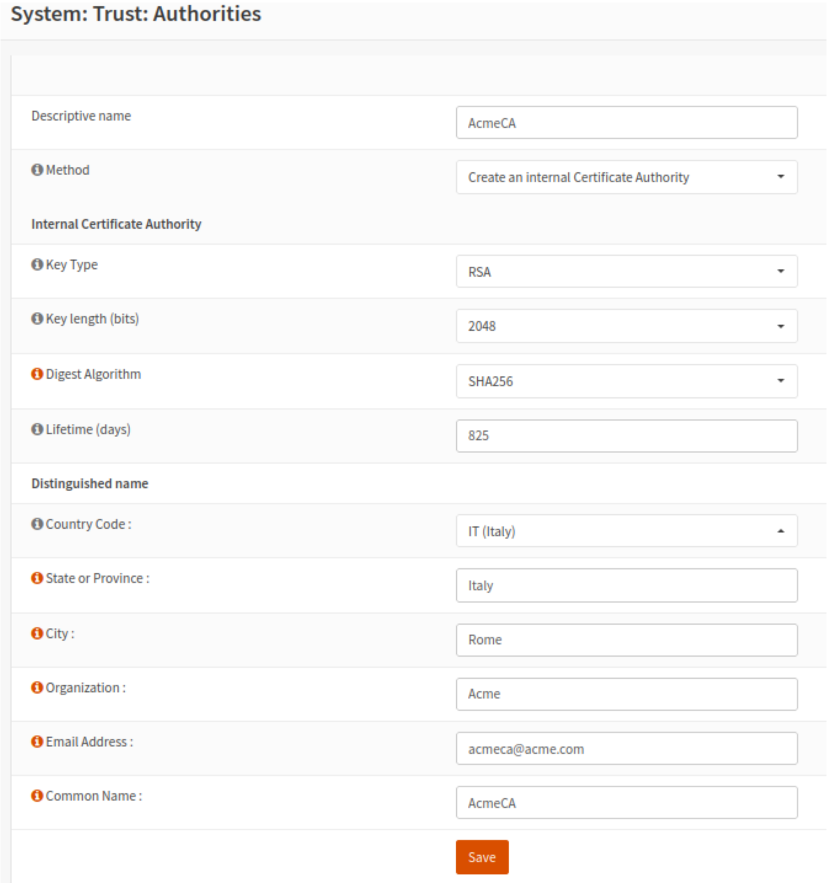
As far as the VPN of the Road-Warriors is concerned, we thought of implementing either two separate VPNs, one for employees and the other for operators, or directly using a single VPN by creating separate subnets or setting static IPs directly. At the end, we chose the second implementation. We tried to set up subnets but without success so we opted for static IPs.

[1] `https://virt-manager.org/`

To understand how to implement OpenVPN, we followed several guides on YouTube[2]and the official OPNsense documentation[3].

## 2.1 Creation of the VPN

Following the guides, we created a new internal Certificate Authority from System → Trust → Authorities. As far as the security values are concerned, we have used the default values RSA 2048, SHA256 hash, and lifetime of 825 days [1].



System: Trust: Authorities

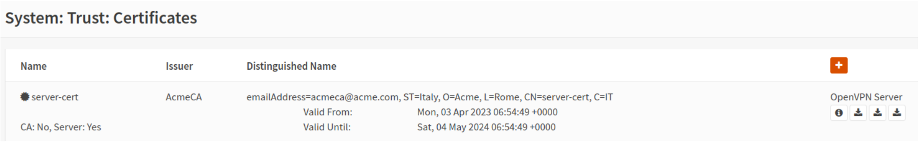
| | |
|---|---|
| Descriptive name | AcmeCA |
| Method | Create an internal Certificate Authority |
| **Internal Certificate Authority** | |
| Key Type | RSA |
| Key length (bits) | 2048 |
| Digest Algorithm | SHA256 |
| Lifetime (days) | 825 |
| **Distinguished name** | |
| Country Code : | IT (Italy) |
| State or Province : | Italy |
| City : | Rome |
| Organization : | Acme |
| Email Address : | acmeca@acme.com |
| Common Name : | AcmeCA |

Figure 1: Creation of the internal Certificate Authority

We generated the internal server's certificate under System → Trust → Certificates from the authority generated earlier [2].

[3] https://www.youtube.com/watch?v=ocGAcZD8qYo,https://www.youtube.com/watch?v=bd0_E6nEFco,https://www.youtube.com/watch?v=QMxjPWDhL2g&t

[3] https://docs.opnsense.org/manual/how-tos/sslvpn_client.html

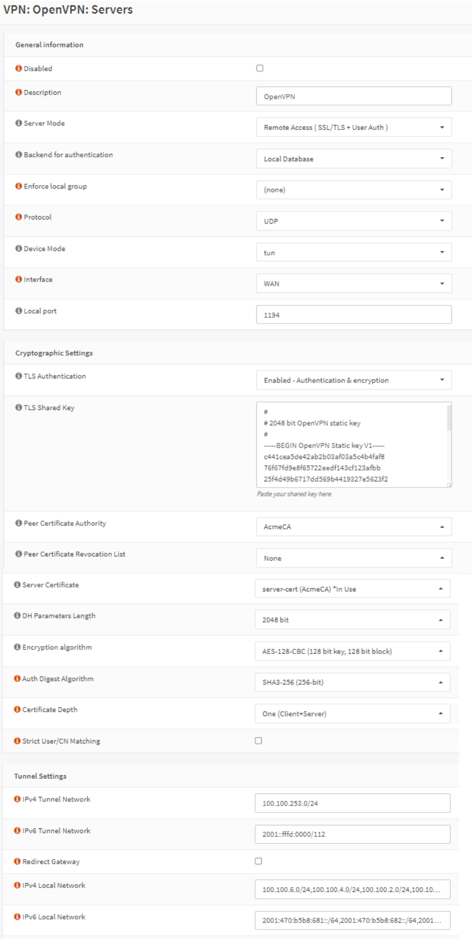


System: Trust: Certificates

| Name | Issuer | Distinguished Name | |
|---|---|---|---|
| server-cert<br>CA: No, Server: Yes | AcmeCA | emailAddress=acmeca@acme.com, ST=Italy, O=Acme, L=Rome, CN=server-cert, C=IT<br>Valid From: Mon, 03 Apr 2023 06:54:49 +0000<br>Valid Until: Sat, 04 May 2024 06:54:49 +0000 | OpenVPN Server |

Figure 2: Creation of the internal Server Certificate

We created the OpenVPN server under VPN → OpenVPN → Servers. The following image shows the chosen parameters [3].



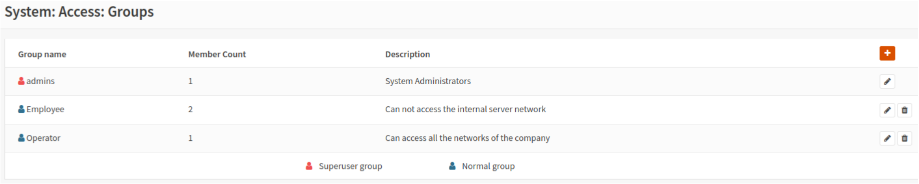
VPN: OpenVPN: Servers

| | |
|---|---|
| **General information** | |
| Disabled | □ |
| Description | OpenVPN |
| Server Mode | Remote Access ( SSL/TLS + User Auth ) |
| Backend for authentication | Local Database |
| Enforce local group | (none) |
| Protocol | UDP |
| Device Mode | tun |
| Interface | WAN |
| Local port | 1194 |
| **Cryptographic Settings** | |
| TLS Authentication | Enabled - Authentication & encryption |
| TLS Shared Key | #<br># 2048 bit OpenVPN static key<br>#<br>-----BEGIN OpenVPN Static key V1-----<br>c441cea5de42ab2b03af03a5c4b4faf8<br>76f67fd9e8f65722eedf143cf123afbb<br>25f4d49b6717dd569b4419327e5623f2<br>*Paste your shared key here.* |
| Peer Certificate Authority | AcmeCA |
| Peer Certificate Revocation List | None |
| Server Certificate | server-cert (AcmeCA) *In Use |
| DH Parameters Length | 2048 bit |
| Encryption algorithm | AES-128-CBC (128 bit key, 128 bit block) |
| Auth Digest Algorithm | SHA3-256 (256-bit) |
| Certificate Depth | One (Client+Server) |
| Strict User/CN Matching | □ |
| **Tunnel Settings** | |
| IPv4 Tunnel Network | 100.100.253.0/24 |
| IPv6 Tunnel Network | 2001::fffd:0000/112 |
| Redirect Gateway | □ |
| IPv4 Local Network | 100.100.6.0/24,100.100.4.0/24,100.100.2.0/24,100.10… |
| IPv6 Local Network | 2001:470:b5b8:681::/64,2001:470:b5b8:682::/64,2001… |

Figure 3: The parameters selected for OpenVPN

As for the remaining values, they were left by default and therefore unchecked except for Dynamic IP and address pool.

Next we generated the two required groups **Operator** for Alice and **Employee** for Bob and Charles (System → Access → Groups) [4].



System: Access: Groups

| Group name | Member Count | Description |
|---|---|---|
| admins | 1 | System Administrators |
| Employee | 2 | Can not access the internal server network |
| Operator | 1 | Can access all the networks of the company |

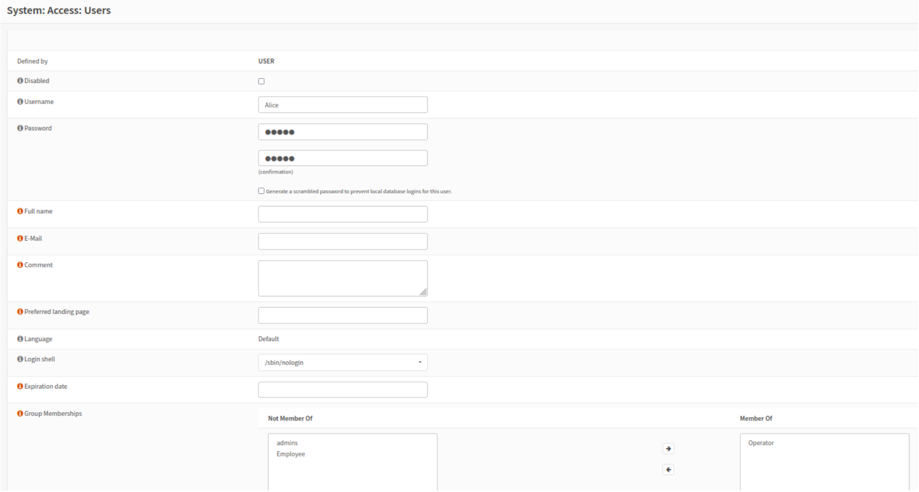
Superuser group    Normal group

Figure 4: Operator and Employee groups

Then we created the three required users Alice, Bob and Charles and added them in the specified groups (System → Access → Users). We have chosen complex and long passwords to make a possible attack more difficult:

```
username: Alice
password: 71xd3b58yodjfB4$I@pY

username: Bob
password: kQ81#&6e#XByuYYM75ME

username: Charles
password: AZ9m4^PQ40vA*PTXF8cR
```

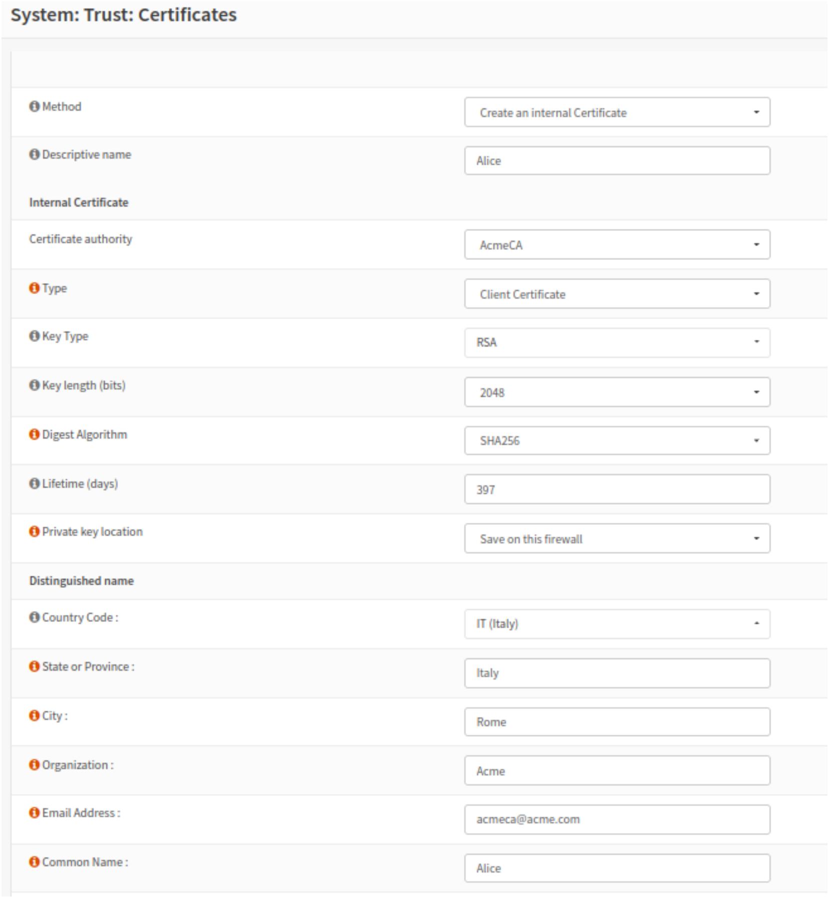
During the creation of the accounts, we generated their certificates using the previous authority (System → Trust → Certificates) Figs. 5 and 6.



System: Access: Users

| | |
|---|---|
| Defined by | USER |
| Disabled | □ |
| Username | Alice |
| Password | ●●●●●<br>●●●●● (confirmation)<br>□ Generate a scrambled password to prevent local database logins for this user. |
| Full name | |
| E-Mail | |
| Comment | |
| Preferred landing page | |
| Language | Default |
| Login shell | /sbin/nologin |
| Expiration date | |
| Group Memberships | Not Member Of: admins, Employee<br>Member Of: Operator |

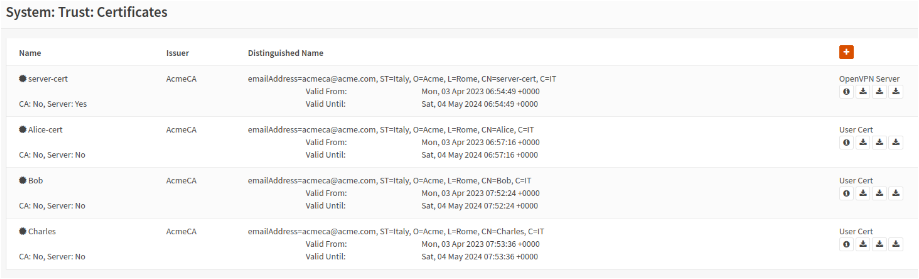
Figure 5: Creation of Alice's account



System: Trust: Certificates

| | |
|---|---|
| Method | Create an internal Certificate |
| Descriptive name | Alice |
| **Internal Certificate** | |
| Certificate authority | AcmeCA |
| Type | Client Certificate |
| Key Type | RSA |
| Key length (bits) | 2048 |
| Digest Algorithm | SHA256 |
| Lifetime (days) | 397 |
| Private key location | Save on this firewall |
| **Distinguished name** | |
| Country Code : | IT (Italy) |
| State or Province : | Italy |
| City : | Rome |
| Organization : | Acme |
| Email Address : | acmeca@acme.com |
| Common Name : | Alice |

Figure 6: Certificate generation for Alice



System: Trust: Certificates

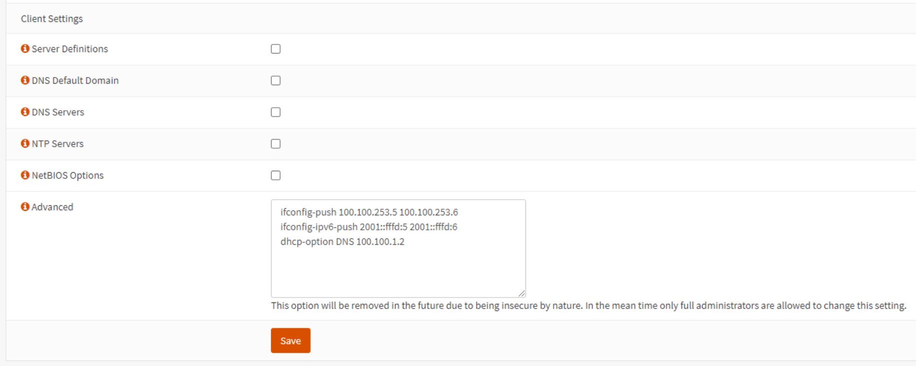
| Name | Issuer | Distinguished Name | + |
|---|---|---|---|
| server-cert<br>CA: No, Server: Yes | AcmeCA | emailAddress=acmeca@acme.com, ST=Italy, O=Acme, L=Rome, CN=server-cert, C=IT<br>Valid From: Mon, 03 Apr 2023 06:54:49 +0000<br>Valid Until: Sat, 04 May 2024 06:54:49 +0000 | OpenVPN Server |
| Alice-cert<br>CA: No, Server: No | AcmeCA | emailAddress=acmeca@acme.com, ST=Italy, O=Acme, L=Rome, CN=Alice, C=IT<br>Valid From: Mon, 03 Apr 2023 06:57:16 +0000<br>Valid Until: Sat, 04 May 2024 06:57:16 +0000 | User Cert |
| Bob<br>CA: No, Server: No | AcmeCA | emailAddress=acmeca@acme.com, ST=Italy, O=Acme, L=Rome, CN=Bob, C=IT<br>Valid From: Mon, 03 Apr 2023 07:52:24 +0000<br>Valid Until: Sat, 04 May 2024 07:52:24 +0000 | User Cert |
| Charles<br>CA: No, Server: No | AcmeCA | emailAddress=acmeca@acme.com, ST=Italy, O=Acme, L=Rome, CN=Charles, C=IT<br>Valid From: Mon, 03 Apr 2023 07:53:36 +0000<br>Valid Until: Sat, 04 May 2024 07:53:36 +0000 | User Cert |

Figure 7: The three generated certificates

To add static IPs to the three users, we added the command *ifconfig-push* for

IPv4 and *ifconfig-ipv6-push* for IPV6 in VPN → OpenVPN → Client Specific Overrides → Advanced [8]. We have chosen IPs:

- Alice: 100.100.253.5 and 2001::fffd:5
- Bob: 100.100.253.9 and 2001::fffd:9
- Charles: 100.100.253.13 and 2001::fffd:13



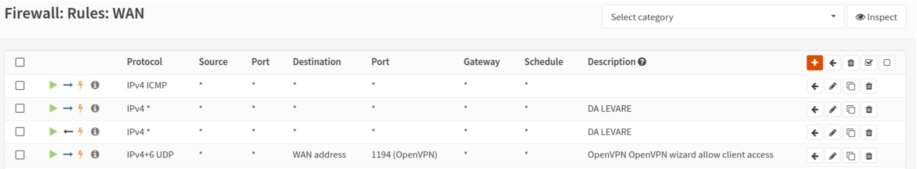
| Client Settings | |
|---|---|
| Server Definitions | □ |
| DNS Default Domain | □ |
| DNS Servers | □ |
| NTP Servers | □ |
| NetBIOS Options | □ |
| Advanced | ifconfig-push 100.100.253.5 100.100.253.6<br>ifconfig-ipv6-push 2001::fffd:5 2001::fffd:6<br>dhcp-option DNS 100.100.1.2<br>This option will be removed in the future due to being insecure by nature. In the mean time only full administrators are allowed to change this setting. |
| | Save |

Figure 8: Commands used to set static IPs in the VPN tunnel for Alice

The same procedure was carried out for the remaining two users.

## 2.2 Firewall rules

Regarding the main router firewall rules, we added a rule in the WAN to allow OpenVPN port 1194 [9]. We left the rules with the description ”DA LEVARE” to allow direct access from the VPN given by the professor to the two routers for the next assignments.

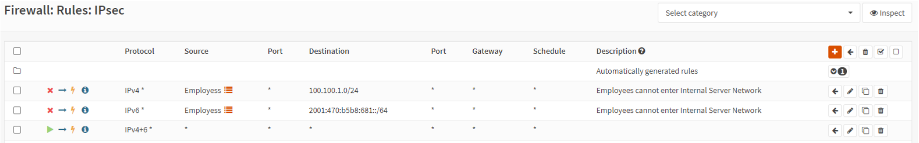


Firewall: Rules: WAN

| □ | | Protocol | Source | Port | Destination | Port | Gateway | Schedule | Description |
|---|---|---|---|---|---|---|---|---|---|
| □ | ▶ → | IPv4 ICMP | * | * | * | * | * | * | |
| □ | ▶ → | IPv4 * | * | * | * | * | * | * | DA LEVARE |
| □ | ▶ ← | IPv4 * | * | * | * | * | * | * | DA LEVARE |
| □ | ▶ → | IPv4+6 UDP | * | * | WAN address | 1194 (OpenVPN) | * | * | OpenVPN OpenVPN wizard allow client access |

Figure 9: Rule to make OpenVPN work

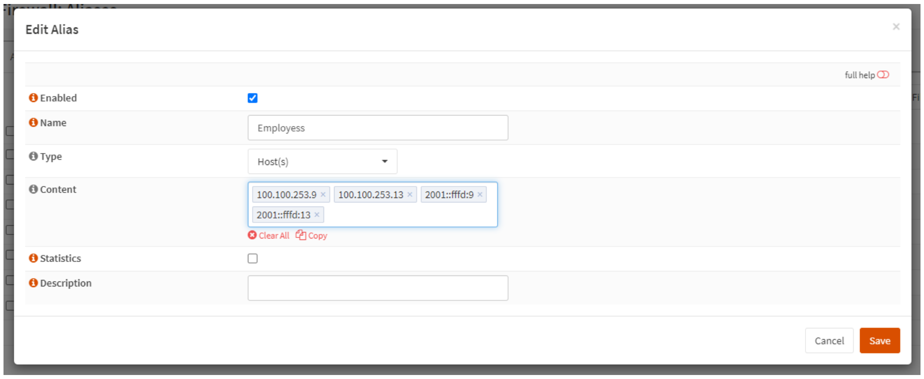
We also added a rule in the internal router firewall to block access to employees to internal servers [10]. This was made possible by the creation of a new alias ”Employess” that contains the IPs of the employees (Firewall → Alliases) [11].



Firewall: Rules: IPsec

| | | Protocol | Source | Port | Destination | Port | Gateway | Schedule | Description |
|---|---|---|---|---|---|---|---|---|---|
| □ | | | | | | | | | Automatically generated rules |
| □ | ✖ → ϟ | IPv4 * | Employess | * | 100.100.1.0/24 | * | * | * | Employees cannot enter Internal Server Network |
| □ | ✖ → ϟ | IPv6 * | Employess | * | 2001:470:b5b8:681::/64 | * | * | * | Employees cannot enter Internal Server Network |
| □ | ▶ → ϟ | IPv4+6 * | * | * | * | * | * | * | |

Figure 10: Rule to block internal network access to Employees in Internal Firewall



Edit Alias

Enabled: ☑

Name: Employess

Type: Host(s)

Content: 100.100.253.9 × 100.100.253.13 × 2001::fffd:9 × 2001::fffd:13 ×

Statistics: □

Description:

Figure 11: Creation of the alias that encapsulates the IPs of the employees

The exported firewall configuration file in the zip was made after completing the second assignment as well, since they were carried out one after the other. For this reason, it contains modifications and the additional rules of the second one.

# 3 Main-Internal VPN tunnel

The internal VPN between the main and the internal router was the first one we implemented. We followed the official OPNsense documentation. Previously we implemented the tunnel with the site-to-site option[4], but by doing this, we encountered some difficulties in getting all packets to communicate through the IPsec tunnel. So in the following we opted for the route based implementation[5].

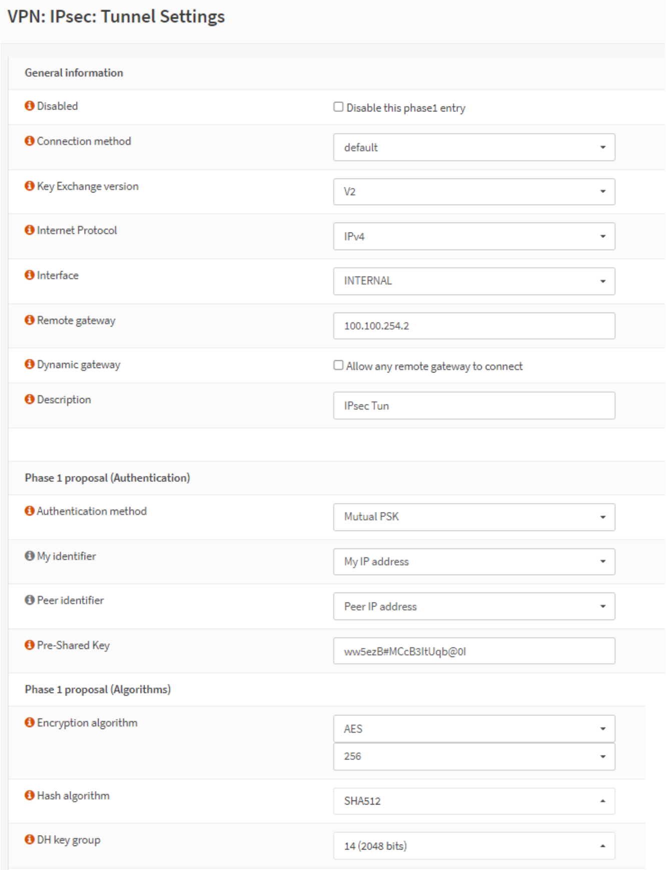
## 3.1 IPsec configuration

In VPN → IPsec → Tunnel Settings of both routers we implemented the two phases of IPsec. In the images below, we show only the implementation on the

[5] https://docs.opnsense.org/manual/how-tos/ipsec-s2s.html

[5] https://docs.opnsense.org/manual/how-tos/ipsec-s2s-route.html

main router, as that on the internal is identical with the IPs inverted and the different Interface. We set a long and complex pre-shared key (*ww5ezB#MCcB3ItUqb@0I*) to make attacks more difficult.



VPN: IPsec: Tunnel Settings

General information

| | |
|---|---|
| Disabled | □ Disable this phase1 entry |
| Connection method | default |
| Key Exchange version | V2 |
| Internet Protocol | IPv4 |
| Interface | INTERNAL |
| Remote gateway | 100.100.254.2 |
| Dynamic gateway | □ Allow any remote gateway to connect |
| Description | IPsec Tun |

Phase 1 proposal (Authentication)

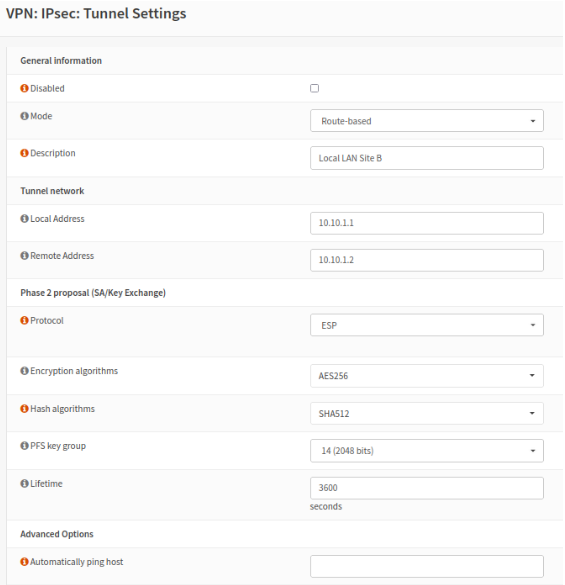
| | |
|---|---|
| Authentication method | Mutual PSK |
| My identifier | My IP address |
| Peer identifier | Peer IP address |
| Pre-Shared Key | ww5ezB#MCcB3ItUqb@0I |

Phase 1 proposal (Algorithms)

| | |
|---|---|
| Encryption algorithm | AES<br>256 |
| Hash algorithm | SHA512 |
| DH key group | 14 (2048 bits) |

Figure 12: IPsec Phase 1 on main firewall

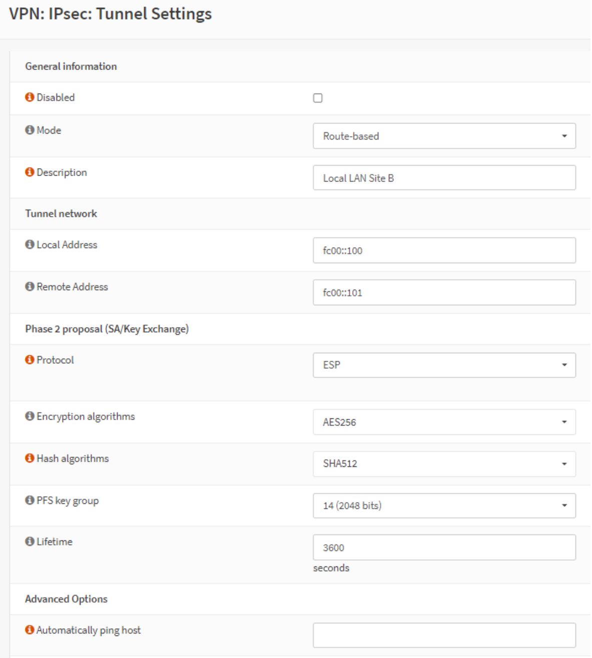
The other options are identical to those in the documentation.



VPN: IPsec: Tunnel Settings

| | |
|---|---|
| **General information** | |
| Disabled | □ |
| Mode | Route-based |
| Description | Local LAN Site B |
| **Tunnel network** | |
| Local Address | 10.10.1.1 |
| Remote Address | 10.10.1.2 |
| **Phase 2 proposal (SA/Key Exchange)** | |
| Protocol | ESP |
| Encryption algorithms | AES256 |
| Hash algorithms | SHA512 |
| PFS key group | 14 (2048 bits) |
| Lifetime | 3600<br>seconds |
| **Advanced Options** | |
| Automatically ping host | |

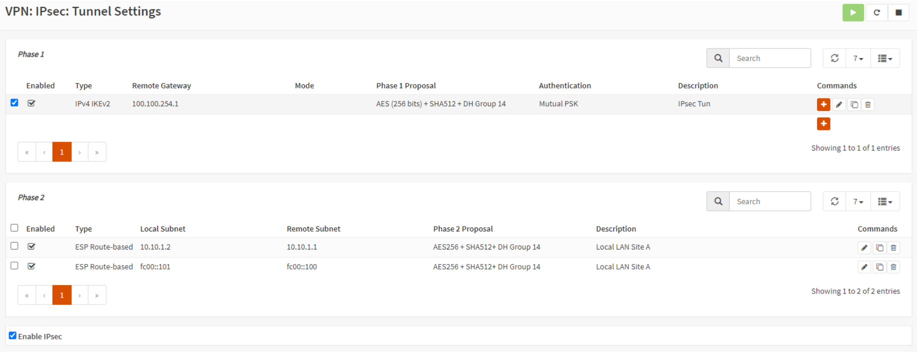
Figure 13: IPsec IPv4 Phase 2 on main firewall



VPN: IPsec: Tunnel Settings

| | |
|---|---|
| **General information** | |
| Disabled | □ |
| Mode | Route-based |
| Description | Local LAN Site B |
| **Tunnel network** | |
| Local Address | fc00::100 |
| Remote Address | fc00::101 |
| **Phase 2 proposal (SA/Key Exchange)** | |
| Protocol | ESP |
| Encryption algorithms | AES256 |
| Hash algorithms | SHA512 |
| PFS key group | 14 (2048 bits) |
| Lifetime | 3600<br>seconds |
| **Advanced Options** | |
| Automatically ping host | |

Figure 14: IPsec IPv6 Phase 2 on main firewall



VPN: IPsec: Tunnel Settings

*Phase 1*

|  | Enabled | Type | Remote Gateway | Mode | Phase 1 Proposal | Authentication | Description | Commands |
|---|---|---|---|---|---|---|---|---|
| ☑ | ☑ | IPv4 IKEv2 | 100.100.254.1 |  | AES (256 bits) + SHA512 + DH Group 14 | Mutual PSK | IPsec Tun |  |

Showing 1 to 1 of 1 entries

*Phase 2*

| □ | Enabled | Type | Local Subnet | Remote Subnet | Phase 2 Proposal | Description | Commands |
|---|---|---|---|---|---|---|---|
| □ | ☑ | ESP Route-based | 10.10.1.2 | 10.10.1.1 | AES256 + SHA512+ DH Group 14 | Local LAN Site A |  |
| □ | ☑ | ESP Route-based | fc00::101 | fc00::100 | AES256 + SHA512+ DH Group 14 | Local LAN Site A |  |

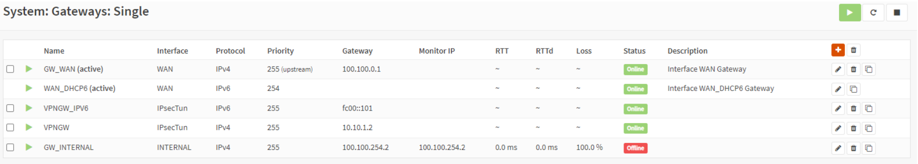
Showing 1 to 2 of 2 entries

☑ Enable IPsec

Figure 15: Implementation on the internal firewall

## 3.2 Changes to gateways and routes

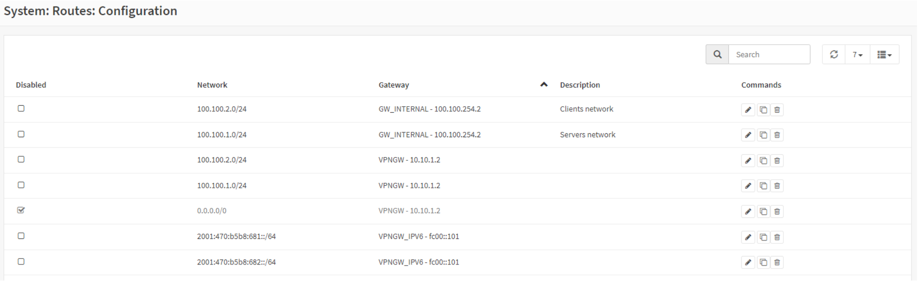
In order to make the route-based setup work, we also modified the gateways and routes of both the Main and Internal routers as stated in the documentation (System → Gateways → Single and System → Routes → Configuration).



System: Gateways: Single

|  |  | Name | Interface | Protocol | Priority | Gateway | Monitor IP | RTT | RTTd | Loss | Status | Description |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| □ | ▶ | GW_WAN (active) | WAN | IPv4 | 255 (upstream) | 100.100.0.1 |  | ~ | ~ | ~ | Online | Interface WAN Gateway |
|  | ▶ | WAN_DHCP6 (active) | WAN | IPv6 | 254 |  |  | ~ | ~ | ~ | Online | Interface WAN_DHCP6 Gateway |
| □ | ▶ | VPNGW_IPV6 | IPsecTun | IPv6 | 255 | fc00::101 |  | ~ | ~ | ~ | Online |  |
| □ | ▶ | VPNGW | IPsecTun | IPv4 | 255 | 10.10.1.2 |  | ~ | ~ | ~ | Online |  |
| □ | ▶ | GW_INTERNAL | INTERNAL | IPv4 | 255 | 100.100.254.2 | 100.100.254.2 | 0.0 ms | 0.0 ms | 100.0 % | Offline |  |

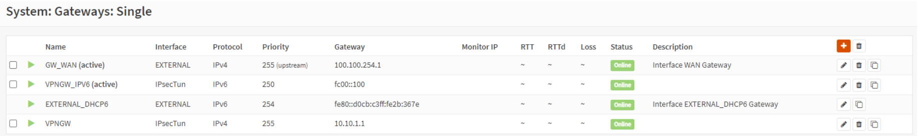
Figure 16: Gateway main router



System: Routes: Configuration

| Disabled | Network | Gateway | Description | Commands |
|---|---|---|---|---|
| □ | 100.100.2.0/24 | GW_INTERNAL - 100.100.254.2 | Clients network |  |
| □ | 100.100.1.0/24 | GW_INTERNAL - 100.100.254.2 | Servers network |  |
| □ | 100.100.2.0/24 | VPNGW - 10.10.1.2 |  |  |
| □ | 100.100.1.0/24 | VPNGW - 10.10.1.2 |  |  |
| ☑ | 0.0.0.0/0 | VPNGW - 10.10.1.2 |  |  |
| □ | 2001:470:b5b8:681::/64 | VPNGW_IPV6 - fc00::101 |  |  |
| □ | 2001:470:b5b8:682::/64 | VPNGW_IPV6 - fc00::101 |  |  |

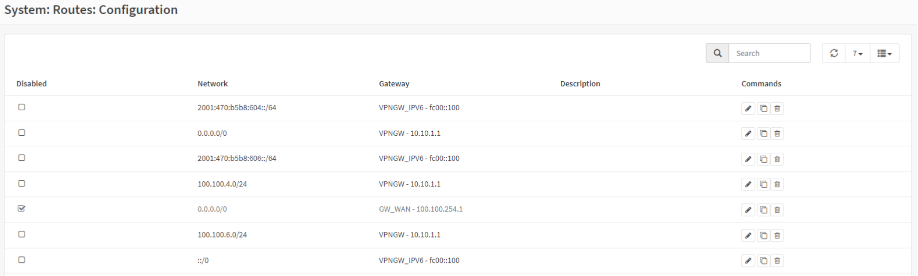
Figure 17: Routes main router (one is disabled)



System: Gateways: Single

| | | Name | Interface | Protocol | Priority | Gateway | Monitor IP | RTT | RTTd | Loss | Status | Description |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| □ | ▶ | GW_WAN (active) | EXTERNAL | IPv4 | 255 (upstream) | 100.100.254.1 | | ~ | ~ | ~ | Online | Interface WAN Gateway |
| □ | ▶ | VPNGW_IPV6 (active) | IPsecTun | IPv6 | 250 | fc00::100 | | ~ | ~ | ~ | Online | |
| | ▶ | EXTERNAL_DHCP6 | EXTERNAL | IPv6 | 254 | fe80::d0cb:c3ff:fe2b:367e | | ~ | ~ | ~ | Online | Interface EXTERNAL_DHCP6 Gateway |
| □ | ▶ | VPNGW | IPsecTun | IPv4 | 255 | 10.10.1.1 | | ~ | ~ | ~ | Online | |

Figure 18: Gateway internal router



System: Routes: Configuration

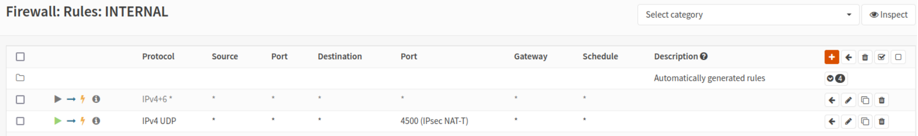
Search 7

| Disabled | Network | Gateway | Description | Commands |
|---|---|---|---|---|
| □ | 2001:470:b5b8:604::/64 | VPNGW_IPV6 - fc00::100 | | |
| □ | 0.0.0.0/0 | VPNGW - 10.10.1.1 | | |
| □ | 2001:470:b5b8:606::/64 | VPNGW_IPV6 - fc00::100 | | |
| □ | 100.100.4.0/24 | VPNGW - 10.10.1.1 | | |
| ☑ | 0.0.0.0/0 | GW_WAN - 100.100.254.1 | | |
| □ | 100.100.6.0/24 | VPNGW - 10.10.1.1 | | |
| □ | ::/0 | VPNGW_IPV6 - fc00::100 | | |

Figure 19: Routes internal router (one is disabled)

## 3.3 Firewall rules

With regard to firewalls, in the main router for the internal interface, we allowed packets to pass through port 4500, which is necessary for IPsec to function. Port 500 and ESP packets were added automatically by OPNsense [20].



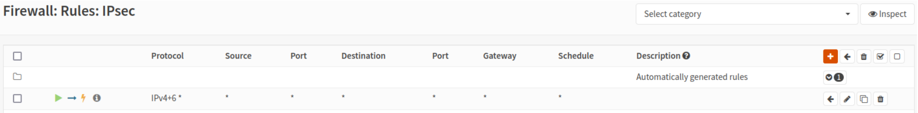
Firewall: Rules: INTERNAL

Select category Inspect

| □ | | Protocol | Source | Port | Destination | Port | Gateway | Schedule | Description |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | Automatically generated rules |
| □ | ▶ → ϟ ℹ | IPv4+6 * | * | * | * | * | * | * | |
| □ | ▶ → ϟ ℹ | IPv4 UDP | * | * | * | 4500 (IPsec NAT-T) | * | * | |

Figure 20: Main router internal interface rules

For the IPsec interface, we allowed all incoming packets to pass through [21].



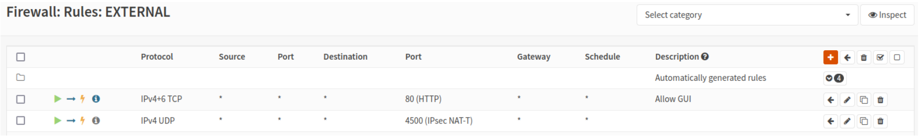
Firewall: Rules: IPsec

Select category Inspect

| □ | | Protocol | Source | Port | Destination | Port | Gateway | Schedule | Description |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | Automatically generated rules |
| □ | ▶ → ϟ ℹ | IPv4+6 * | * | * | * | * | * | * | |

Figure 21: Main router IPsec interface rules

Regarding the internal router firewall for the External interface, we added two rules: one for web access to the internal router and the other for port 4500 [22] just like how we did previously.

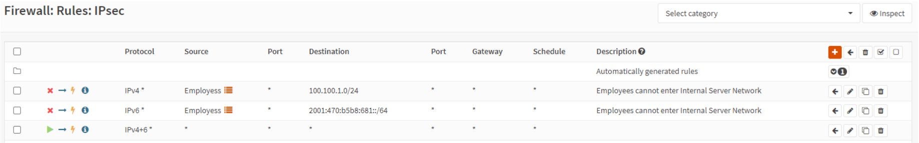


Firewall: Rules: EXTERNAL

| | Protocol | Source | Port | Destination | Port | Gateway | Schedule | Description |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | Automatically generated rules |
| | IPv4+6 TCP | * | * | * | 80 (HTTP) | * | * | Allow GUI |
| | IPv4 UDP | * | * | * | 4500 (IPsec NAT-T) | * | * | |

Figure 22: Internal router external interface rules

For the IPsec interface, we first blocked access to the employees and then allowed access to the others [23].



Firewall: Rules: IPsec

| | Protocol | Source | Port | Destination | Port | Gateway | Schedule | Description |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | Automatically generated rules |
| ✗ | IPv4 * | Employess | * | 100.100.1.0/24 | * | * | * | Employees cannot enter Internal Server Network |
| ✗ | IPv6 * | Employess | * | 2001:470:b5b8:681::/64 | * | * | * | Employees cannot enter Internal Server Network |
| ▶ | IPv4+6 * | * | * | * | * | * | * | |

Figure 23: Internal router IPsec interface rules

The exported firewall configuration file in the zip was made after completing the second assignment as well, since they were carried out one after the other. For this reason, it contains modifications and the additional rules of the second one.

# 4 IPv6 Implementation

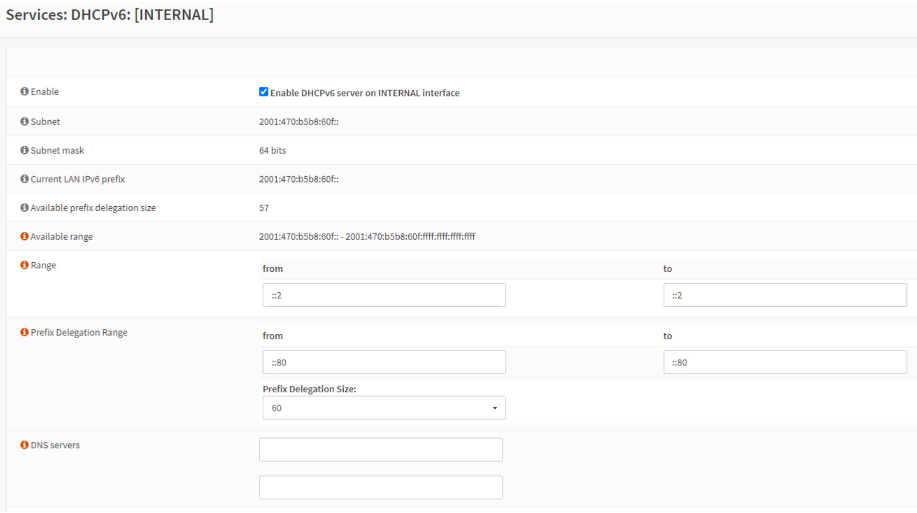
We also decided to implement IPv6. For the implementation we followed the video of the professor so we implemented it through SLAAC and DHCPv6-PD.

We were unable to implement IPv6 on the *fanstasticcofee* and *Greenbone* machines. With the first one we could not connect while with the second we had problems running the command from root.

## 4.1 Steps

From the main firewall (in Interfaces → [WAN]) we set the IPv6 Configuration Type to DHCPv6. By doing this, we requested from the ISP an IPv6 prefix. We set the prefix delegation size to 56 as stated by the professor.

Subsequently, in the EXTERNAL_CLIENT, DMZ and INTERNAL interfaces we set IPv6 Configuration Type to Track Interface and the IPv6 Prefix ID respectively to 4, 6 and f. For the INTERNAL interface, to provide IPv6 to the internal firewall, we also checked the option Manual configuration. By doing this, in Services → Router Advertisements we could set the option for the INTERNAL interface. The only option that we changed was to put Router Advertisements on Managed. Then we enabled DHCPv6 in Services → DHCPv6 → [INTERNAL].



Services: DHCPv6: [INTERNAL]

| | |
|---|---|
| Enable | ☑ Enable DHCPv6 server on INTERNAL interface |
| Subnet | 2001:470:b5b8:60f:: |
| Subnet mask | 64 bits |
| Current LAN IPv6 prefix | 2001:470:b5b8:60f:: |
| Available prefix delegation size | 57 |
| Available range | 2001:470:b5b8:60f:: - 2001:470:b5b8:60f:ffff:ffff:ffff:ffff |
| Range | from ::2 to ::2 |
| Prefix Delegation Range | from ::80 to ::80<br>Prefix Delegation Size: 60 |
| DNS servers | |

Figure 24: Chosen option for DHCPv6

Thanks to this setting we were able to send to the internal firewall an IPv6 prefix. This has been done in Interfaces → [EXTERNAL]: we configured IPv6 Configuration Type to DHCPv6 and we used 60 as Prefix delegation size.

Just like we did previously, we had to configure the internal firewall interfaces (CLIENTS and SERVERS) to accept IPv6 in Track Interface mode and we used as prefix 2 and 1.

In this way we created different subnets:

- Internal servers network: 2001:470:b5b8:681::/64
- Clients network: 2001:470:b5b8:682::/64
- External services: 2001:470:b5b8:604::/64
- DMZ: 2001:470:b5b8:606::/64

As stated before, all the hosts get the IP thanks to SLAAC. We configured wherever it was possible **stable-privacy**:

- client-ext-1: 2001:470:b5b8:604:5f6:e997:b8fd:6e90
- kali: 2001:470:b5b8:682:ef01:a2e3:bf49:247

- arpwatch: 2001:470:b5b8:682:d546:21c4:562d:a82b
- dnsserver: 2001:470:b5b8:681:6bbe:b411:54a3:2a93
- logserver: 2001:470:b5b8:681:4060:2447:8e45:eebd
- webserver: 2001:470:b5b8:606:8033:256e:38b7:19e9
- proxyserver: 2001:470:b5b8:606:3037:177c:bc7c:aabf

To implement it we modified in every host the configuration file */etc/sysctl.d/99-sysctl.conf* by adding the lines:

```
net.ipv6.conf.eth0.stable_secret = X:X:X:X:X:X:X:X
net.ipv6.conf.eth0.addr_gen_mode=3
```

The value marked as X where assigned randomly. For the other host we left the default configuration with EUI-64.

### 4.2 IPv6 IPsec

To make **IPsec** work in IPv6 we also had to add an IPv6 second phase as can be seen in the images 14 and 15. We have chosen as local subnet and remote subnet unique local adresses. We also modified the gateways and routes of both the routers as previously stated in Figs. 16 to 19 so all the packets between the internal and main routers are only able to go through IPsec.

### 4.3 IPv6 OpenVPN

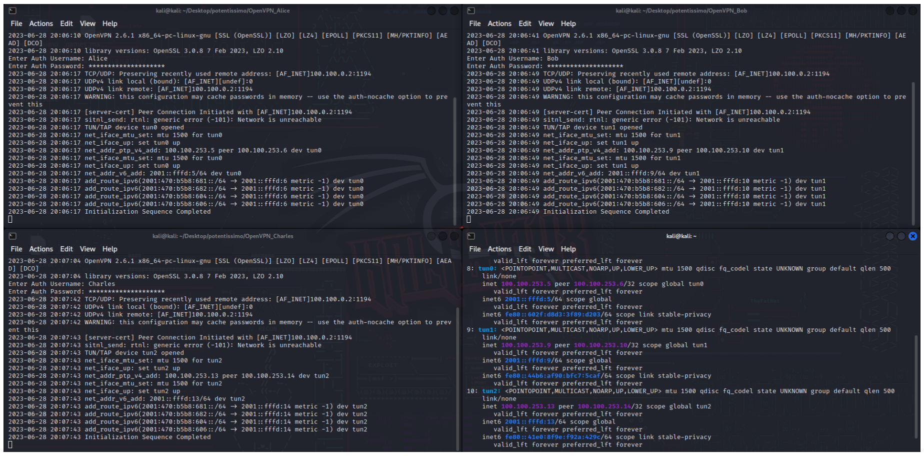
For **OpenVPN** we had to modify the configuration as seen in the image 3, we added static IPv6 to the users (image 8) and modified Employess alias (image 11).

## 5 Test of the new configuration

We performed numerous tests to verify the correct functioning of our implementations.

### 5.1 Road-Warriors VPN tests

With regard to the Road-Warriors' VPN, we downloaded the users' keys from VPN → OpenVPN → Client Export and tested the correct functioning. As shown in the image below, the users are able to connect to the main router giving them the eligibility to receive the static IP addresses assigned [25].


kali@kali: ~/Desktop/potentissimo/OpenVPN_Alice
File Actions Edit View Help
kali@kali: ~/Desktop/potentissimo/OpenVPN_Bob
File Actions Edit View Help
kali@kali: ~/Desktop/potentissimo/OpenVPN_Charles
File Actions Edit View Help
kali@kali: ~
File Actions Edit View Help

```
2023-06-28 20:06:10 OpenVPN 2.6.1 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AE
AD] [DCO]
2023-06-28 20:06:10 library versions: OpenSSL 3.0.8 7 Feb 2023, LZO 2.10
Enter Auth Username: Alice
Enter Auth Password: *******************
2023-06-28 20:06:17 TCP/UDP: Preserving recently used remote address: [AF_INET]100.100.0.2:1194
2023-06-28 20:06:17 UDPv4 link local (bound): [AF_INET][undef]:0
2023-06-28 20:06:17 UDPv4 link remote: [AF_INET]100.100.0.2:1194
2023-06-28 20:06:17 WARNING: this configuration may cache passwords in memory -- use the auth-nocache option to pre
vent this
2023-06-28 20:06:17 [server-cert] Peer Connection Initiated with [AF_INET]100.100.0.2:1194
2023-06-28 20:06:17 sitnl_send: rtnl: generic error (-101): Network is unreachable
2023-06-28 20:06:17 TUN/TAP device tun0 opened
2023-06-28 20:06:17 net_iface_mtu_set: mtu 1500 for tun0
2023-06-28 20:06:17 net_iface_up: set tun0 up
2023-06-28 20:06:17 net_addr_ptp_v4_add: 100.100.253.5 peer 100.100.253.6 dev tun0
2023-06-28 20:06:17 net_iface_mtu_set: mtu 1500 for tun0
2023-06-28 20:06:17 net_iface_up: set tun0 up
2023-06-28 20:06:17 net_addr_v6_add: 2001::fffd:5/64 dev tun0
2023-06-28 20:06:17 add_route_ipv6(2001:470:b5b8:681::/64 → 2001::fffd:6 metric -1) dev tun0
2023-06-28 20:06:17 add_route_ipv6(2001:470:b5b8:682::/64 → 2001::fffd:6 metric -1) dev tun0
2023-06-28 20:06:17 add_route_ipv6(2001:470:b5b8:604::/64 → 2001::fffd:6 metric -1) dev tun0
2023-06-28 20:06:17 add_route_ipv6(2001:470:b5b8:606::/64 → 2001::fffd:6 metric -1) dev tun0
2023-06-28 20:06:17 Initialization Sequence Completed
▯
```

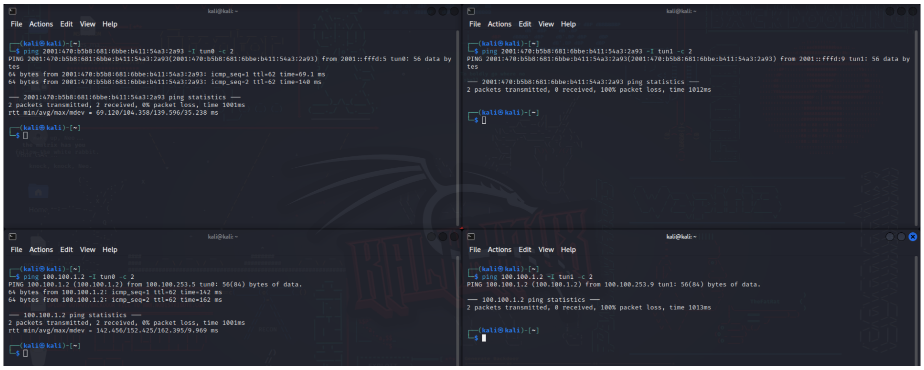
```
2023-06-28 20:06:41 OpenVPN 2.6.1 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AE
AD] [DCO]
2023-06-28 20:06:41 library versions: OpenSSL 3.0.8 7 Feb 2023, LZO 2.10
Enter Auth Username: Bob
Enter Auth Password: *******************
2023-06-28 20:06:49 TCP/UDP: Preserving recently used remote address: [AF_INET]100.100.0.2:1194
2023-06-28 20:06:49 UDPv4 link local (bound): [AF_INET][undef]:0
2023-06-28 20:06:49 UDPv4 link remote: [AF_INET]100.100.0.2:1194
2023-06-28 20:06:49 WARNING: this configuration may cache passwords in memory -- use the auth-nocache option to pre
vent this
2023-06-28 20:06:49 [server-cert] Peer Connection Initiated with [AF_INET]100.100.0.2:1194
2023-06-28 20:06:49 sitnl_send: rtnl: generic error (-101): Network is unreachable
2023-06-28 20:06:49 TUN/TAP device tun1 opened
2023-06-28 20:06:49 net_iface_mtu_set: mtu 1500 for tun1
2023-06-28 20:06:49 net_iface_up: set tun1 up
2023-06-28 20:06:49 net_addr_ptp_v4_add: 100.100.253.9 peer 100.100.253.10 dev tun1
2023-06-28 20:06:49 net_iface_mtu_set: mtu 1500 for tun1
2023-06-28 20:06:49 net_iface_up: set tun1 up
2023-06-28 20:06:49 net_addr_v6_add: 2001::fffd:9/64 dev tun1
2023-06-28 20:06:49 add_route_ipv6(2001:470:b5b8:681::/64 → 2001::fffd:10 metric -1) dev tun1
2023-06-28 20:06:49 add_route_ipv6(2001:470:b5b8:682::/64 → 2001::fffd:10 metric -1) dev tun1
2023-06-28 20:06:49 add_route_ipv6(2001:470:b5b8:604::/64 → 2001::fffd:10 metric -1) dev tun1
2023-06-28 20:06:49 add_route_ipv6(2001:470:b5b8:606::/64 → 2001::fffd:10 metric -1) dev tun1
2023-06-28 20:06:49 Initialization Sequence Completed
▯
```

```
2023-06-28 20:07:04 OpenVPN 2.6.1 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AEA
D] [DCO]
2023-06-28 20:07:04 library versions: OpenSSL 3.0.8 7 Feb 2023, LZO 2.10
Enter Auth Username: Charles
Enter Auth Password: *******************
2023-06-28 20:07:42 TCP/UDP: Preserving recently used remote address: [AF_INET]100.100.0.2:1194
2023-06-28 20:07:42 UDPv4 link local (bound): [AF_INET][undef]:0
2023-06-28 20:07:42 UDPv4 link remote: [AF_INET]100.100.0.2:1194
2023-06-28 20:07:42 WARNING: this configuration may cache passwords in memory -- use the auth-nocache option to prev
ent this
2023-06-28 20:07:43 [server-cert] Peer Connection Initiated with [AF_INET]100.100.0.2:1194
2023-06-28 20:07:43 sitnl_send: rtnl: generic error (-101): Network is unreachable
2023-06-28 20:07:43 TUN/TAP device tun2 opened
2023-06-28 20:07:43 net_iface_mtu_set: mtu 1500 for tun2
2023-06-28 20:07:43 net_iface_up: set tun2 up
2023-06-28 20:07:43 net_addr_ptp_v4_add: 100.100.253.13 peer 100.100.253.14 dev tun2
2023-06-28 20:07:43 net_iface_mtu_set: mtu 1500 for tun2
2023-06-28 20:07:43 net_iface_up: set tun2 up
2023-06-28 20:07:43 net_addr_v6_add: 2001::fffd:13/64 dev tun2
2023-06-28 20:07:43 add_route_ipv6(2001:470:b5b8:681::/64 → 2001::fffd:14 metric -1) dev tun2
2023-06-28 20:07:43 add_route_ipv6(2001:470:b5b8:682::/64 → 2001::fffd:14 metric -1) dev tun2
2023-06-28 20:07:43 add_route_ipv6(2001:470:b5b8:604::/64 → 2001::fffd:14 metric -1) dev tun2
2023-06-28 20:07:43 add_route_ipv6(2001:470:b5b8:606::/64 → 2001::fffd:14 metric -1) dev tun2
2023-06-28 20:07:43 Initialization Sequence Completed
▯
```

```
       valid_lft forever preferred_lft forever
8: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 100.100.253.5 peer 100.100.253.6/32 scope global tun0
       valid_lft forever preferred_lft forever
    inet6 2001::fffd:5/64 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::602f:d8d3:3f89:d203/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
9: tun1: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 100.100.253.9 peer 100.100.253.10/32 scope global tun1
       valid_lft forever preferred_lft forever
    inet6 2001::fffd:9/64 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::44b6:af90:bfc7:5caf/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
10: tun2: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 100.100.253.13 peer 100.100.253.14/32 scope global tun2
       valid_lft forever preferred_lft forever
    inet6 2001::fffd:13/64 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::41e0:8f9e:f92a:429c/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
```

Figure 25: Example of the proper functioning of the VPNs

The following test, on the other hand, shows that we are able to ping the dns server with Alice, whereas with Bob this is not possible since he is an employee and cannot access the internal server subnet. The test was carried out with both IPv4 and IPv6 [26]. We also verified that the packets go through the IPsec tunnel.


kali@kali: ~
File Actions Edit View Help
kali@kali: ~
File Actions Edit View Help
kali@kali: ~
File Actions Edit View Help
kali@kali: ~
File Actions Edit View Help

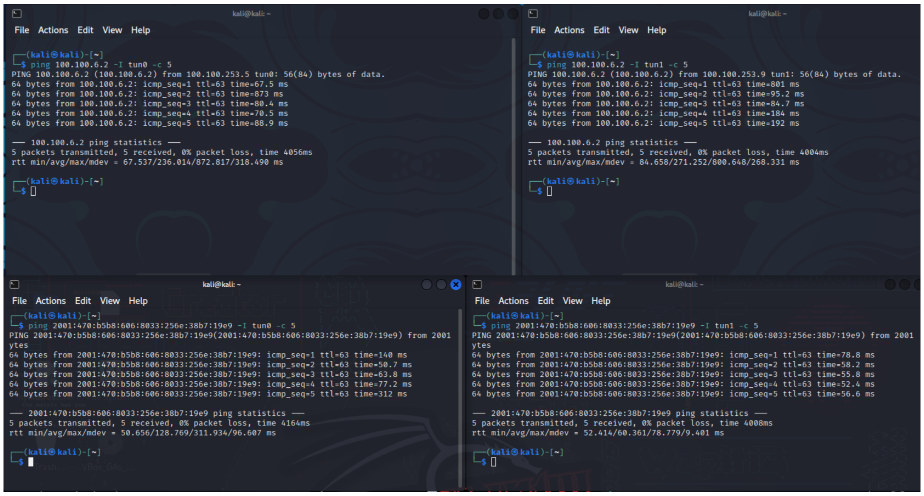
```
┌──(kali㉿kali)-[~]
└─$ ping 2001:470:b5b8:681:6bbe:b411:54a3:2a93 -I tun0 -c 2
PING 2001:470:b5b8:681:6bbe:b411:54a3:2a93(2001:470:b5b8:681:6bbe:b411:54a3:2a93) from 2001::fffd:5 tun0: 56 data by
tes
64 bytes from 2001:470:b5b8:681:6bbe:b411:54a3:2a93: icmp_seq=1 ttl=62 time=69.1 ms
64 bytes from 2001:470:b5b8:681:6bbe:b411:54a3:2a93: icmp_seq=2 ttl=62 time=140 ms

—— 2001:470:b5b8:681:6bbe:b411:54a3:2a93 ping statistics ——
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 69.120/104.358/139.596/35.238 ms

┌──(kali㉿kali)-[~]
└─$ ▯
```

```
┌──(kali㉿kali)-[~]
└─$ ping 2001:470:b5b8:681:6bbe:b411:54a3:2a93 -I tun1 -c 2
PING 2001:470:b5b8:681:6bbe:b411:54a3:2a93(2001:470:b5b8:681:6bbe:b411:54a3:2a93) from 2001::fffd:9 tun1: 56 data by
tes

—— 2001:470:b5b8:681:6bbe:b411:54a3:2a93 ping statistics ——
2 packets transmitted, 0 received, 100% packet loss, time 1012ms


┌──(kali㉿kali)-[~]
└─$ ▯
```

```
┌──(kali㉿kali)-[~]
└─$ ping 100.100.1.2 -I tun0 -c 2
PING 100.100.1.2 (100.100.1.2) from 100.100.253.5 tun0: 56(84) bytes of data.
64 bytes from 100.100.1.2: icmp_seq=1 ttl=62 time=142 ms
64 bytes from 100.100.1.2: icmp_seq=2 ttl=62 time=162 ms

—— 100.100.1.2 ping statistics ——
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 142.456/152.425/162.395/9.969 ms

┌──(kali㉿kali)-[~]
└─$ ▯
```

```
┌──(kali㉿kali)-[~]
└─$ ping 100.100.1.2 -I tun1 -c 2
PING 100.100.1.2 (100.100.1.2) from 100.100.253.9 tun1: 56(84) bytes of data.

—— 100.100.1.2 ping statistics ——
2 packets transmitted, 0 received, 100% packet loss, time 1013ms


┌──(kali㉿kali)-[~]
└─$ ▯
```

Figure 26: Tests to verify the proper functioning of VPNs



```
┌──(kali㉿kali)-[~]
└─$ ping 100.100.6.2 -I tun0 -c 5
PING 100.100.6.2 (100.100.6.2) from 100.100.253.5 tun0: 56(84) bytes of data.
64 bytes from 100.100.6.2: icmp_seq=1 ttl=63 time=67.5 ms
64 bytes from 100.100.6.2: icmp_seq=2 ttl=63 time=873 ms
64 bytes from 100.100.6.2: icmp_seq=3 ttl=63 time=80.4 ms
64 bytes from 100.100.6.2: icmp_seq=4 ttl=63 time=70.5 ms
64 bytes from 100.100.6.2: icmp_seq=5 ttl=63 time=88.9 ms

── 100.100.6.2 ping statistics ──
5 packets transmitted, 5 received, 0% packet loss, time 4056ms
rtt min/avg/max/mdev = 67.537/236.014/872.817/318.490 ms

┌──(kali㉿kali)-[~]
└─$ 
```

```
┌──(kali㉿kali)-[~]
└─$ ping 100.100.6.2 -I tun1 -c 5
PING 100.100.6.2 (100.100.6.2) from 100.100.253.9 tun1: 56(84) bytes of data.
64 bytes from 100.100.6.2: icmp_seq=1 ttl=63 time=801 ms
64 bytes from 100.100.6.2: icmp_seq=2 ttl=63 time=95.2 ms
64 bytes from 100.100.6.2: icmp_seq=3 ttl=63 time=84.7 ms
64 bytes from 100.100.6.2: icmp_seq=4 ttl=63 time=184 ms
64 bytes from 100.100.6.2: icmp_seq=5 ttl=63 time=192 ms

── 100.100.6.2 ping statistics ──
5 packets transmitted, 5 received, 0% packet loss, time 4004ms
rtt min/avg/max/mdev = 84.658/271.252/800.648/268.331 ms

┌──(kali㉿kali)-[~]
└─$ 
```

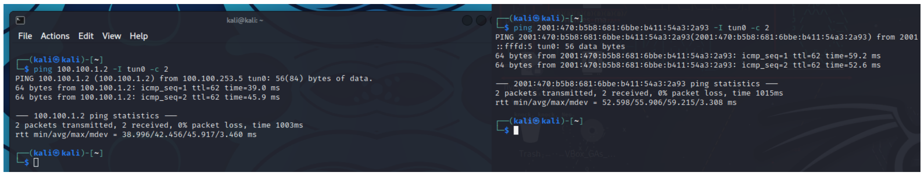
```
┌──(kali㉿kali)-[~]
└─$ ping 2001:470:b5b8:606:8033:256e:38b7:19e9 -I tun0 -c 5
PING 2001:470:b5b8:606:8033:256e:38b7:19e9(2001:470:b5b8:606:8033:256e:38b7:19e9) from 2001
ytes
64 bytes from 2001:470:b5b8:606:8033:256e:38b7:19e9: icmp_seq=1 ttl=63 time=140 ms
64 bytes from 2001:470:b5b8:606:8033:256e:38b7:19e9: icmp_seq=2 ttl=63 time=50.7 ms
64 bytes from 2001:470:b5b8:606:8033:256e:38b7:19e9: icmp_seq=3 ttl=63 time=63.8 ms
64 bytes from 2001:470:b5b8:606:8033:256e:38b7:19e9: icmp_seq=4 ttl=63 time=77.2 ms
64 bytes from 2001:470:b5b8:606:8033:256e:38b7:19e9: icmp_seq=5 ttl=63 time=312 ms

── 2001:470:b5b8:606:8033:256e:38b7:19e9 ping statistics ──
5 packets transmitted, 5 received, 0% packet loss, time 4164ms
rtt min/avg/max/mdev = 50.656/128.769/311.934/96.607 ms

┌──(kali㉿kali)-[~]
└─$ 
```

```
┌──(kali㉿kali)-[~]
└─$ ping 2001:470:b5b8:606:8033:256e:38b7:19e9 -I tun1 -c 5
PING 2001:470:b5b8:606:8033:256e:38b7:19e9(2001:470:b5b8:606:8033:256e:38b7:19e9) from 2001
ytes
64 bytes from 2001:470:b5b8:606:8033:256e:38b7:19e9: icmp_seq=1 ttl=63 time=78.8 ms
64 bytes from 2001:470:b5b8:606:8033:256e:38b7:19e9: icmp_seq=2 ttl=63 time=58.2 ms
64 bytes from 2001:470:b5b8:606:8033:256e:38b7:19e9: icmp_seq=3 ttl=63 time=55.8 ms
64 bytes from 2001:470:b5b8:606:8033:256e:38b7:19e9: icmp_seq=4 ttl=63 time=52.4 ms
64 bytes from 2001:470:b5b8:606:8033:256e:38b7:19e9: icmp_seq=5 ttl=63 time=56.6 ms

── 2001:470:b5b8:606:8033:256e:38b7:19e9 ping statistics ──
5 packets transmitted, 5 received, 0% packet loss, time 4008ms
rtt min/avg/max/mdev = 52.414/60.361/78.779/9.401 ms

┌──(kali㉿kali)-[~]
└─$ 
```

Figure 27: Test to show that both groups are able to ping the web server that is outside the internal server network

## 5.2 Internal VPN tests

In this section, we demonstrate the proper functioning of the IPsec tunnel. The following example shows the ping performed by Alice to the DNS server. As shown on the image below, the packets pass through the IPsec tunnel.

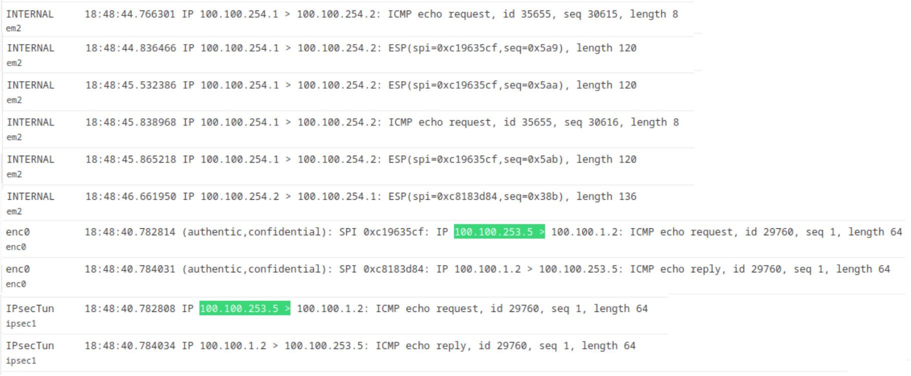


```
┌──(kali㉿kali)-[~]
└─$ ping 100.100.1.2 -I tun0 -c 2
PING 100.100.1.2 (100.100.1.2) from 100.100.253.5 tun0: 56(84) bytes of data.
64 bytes from 100.100.1.2: icmp_seq=1 ttl=62 time=39.0 ms
64 bytes from 100.100.1.2: icmp_seq=2 ttl=62 time=45.9 ms

── 100.100.1.2 ping statistics ──
2 packets transmitted, 2 received, 0% packet loss, time 1003ms
rtt min/avg/max/mdev = 38.996/42.456/45.917/3.460 ms

┌──(kali㉿kali)-[~]
└─$ 
```

```
┌──(kali㉿kali)-[~]
└─$ ping 2001:470:b5b8:681:6bbe:b411:54a3:2a93 -I tun0 -c 2
PING 2001:470:b5b8:681:6bbe:b411:54a3:2a93(2001:470:b5b8:681:6bbe:b411:54a3:2a93) from 2001
::fffd:5 tun0: 56 data bytes
64 bytes from 2001:470:b5b8:681:6bbe:b411:54a3:2a93: icmp_seq=1 ttl=62 time=59.2 ms
64 bytes from 2001:470:b5b8:681:6bbe:b411:54a3:2a93: icmp_seq=2 ttl=62 time=52.6 ms

── 2001:470:b5b8:681:6bbe:b411:54a3:2a93 ping statistics ──
2 packets transmitted, 2 received, 0% packet loss, time 1015ms
rtt min/avg/max/mdev = 52.598/55.906/59.215/3.308 ms

┌──(kali㉿kali)-[~]
└─$ 
```

Figure 28: Test to show proper functioning of IPsec

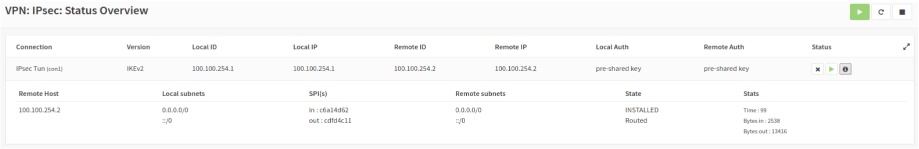
Packets were captured by the main router on the INTERNAL, IPsecTun and IPsec interfaces using the PacketCapture tool (Interfaces → Diagnostic → PacketCapture). On the INTERNAL interface, we can see that ESP packets are exchanged, while on the IPsec interface, pings between hosts can be seen clearly.



```
INTERNAL      18:48:44.766301 IP 100.100.254.1 > 100.100.254.2: ICMP echo request, id 35655, seq 30615, length 8
em2
INTERNAL      18:48:44.836466 IP 100.100.254.1 > 100.100.254.2: ESP(spi=0xc19635cf,seq=0x5a9), length 120
em2
INTERNAL      18:48:45.532386 IP 100.100.254.1 > 100.100.254.2: ESP(spi=0xc19635cf,seq=0x5aa), length 120
em2
INTERNAL      18:48:45.838968 IP 100.100.254.1 > 100.100.254.2: ICMP echo request, id 35655, seq 30616, length 8
em2
INTERNAL      18:48:45.865218 IP 100.100.254.1 > 100.100.254.2: ESP(spi=0xc19635cf,seq=0x5ab), length 120
em2
INTERNAL      18:48:46.661950 IP 100.100.254.2 > 100.100.254.1: ESP(spi=0xc8183d84,seq=0x38b), length 136
em2
enc0          18:48:40.782814 (authentic,confidential): SPI 0xc19635cf: IP 100.100.253.5 > 100.100.1.2: ICMP echo request, id 29760, seq 1, length 64
enc0
enc0          18:48:40.784031 (authentic,confidential): SPI 0xc8183d84: IP 100.100.1.2 > 100.100.253.5: ICMP echo reply, id 29760, seq 1, length 64
enc0
IPsecTun      18:48:40.782808 IP 100.100.253.5 > 100.100.1.2: ICMP echo request, id 29760, seq 1, length 64
ipsec1
IPsecTun      18:48:40.784034 IP 100.100.1.2 > 100.100.253.5: ICMP echo reply, id 29760, seq 1, length 64
ipsec1
```

Figure 29: IPv4 packets captured by the main router, the same can be done for IPv6

It is also possible to view the correct functioning in VPN → IPsec → Status Overview with the increase in bytes as the pings pass.



VPN: IPsec: Status Overview

| Connection | Version | Local ID | Local IP | Remote ID | Remote IP | Local Auth | Remote Auth | Status |
|---|---|---|---|---|---|---|---|---|
| IPsec Tun (con1) | IKEv2 | 100.100.254.1 | 100.100.254.1 | 100.100.254.2 | 100.100.254.2 | pre-shared key | pre-shared key | ✖ ▶ ℹ |

| Remote Host | Local subnets | SPI(s) | Remote subnets | State | Stats |
|---|---|---|---|---|---|
| 100.100.254.2 | 0.0.0.0/0<br>::/0 | in : c6a14d62<br>out : cdfd4c11 | 0.0.0.0/0<br>::/0 | INSTALLED<br>Routed | Time : 99<br>Bytes in : 2538<br>Bytes out : 13416 |

Figure 30: Demonstration of the increase in bytes

# 6 Final remarks

Thanks to this assignment, we learnt how to use OPNsense and set up VPNs on it. The implementation of the functionality took us a long time because every time we managed to deploy more features, we caused damage to the previous ones. Very often we had to reset the hosts and start the implementation all over again. Fortunately, however, once we learnt how to implement it, doing it again was easy and repeatable in a short amount of time. The negative factors we found in OPNsense were that there are not many documentations or sources of information on its use and that it is often difficult to understand the cause of errors.