Practical Network Defense

ASSIGNMENT 2

VPN on ACME co.

Group number: **11**

Andrea Fede - 1942562
Chiara Iannicelli - 1957045
Pietro Colaguori - 1936709
Riccardo Tuzzolino - 1954109

Contents

1 Initial Brainstorming

The assignment required to configure two new services running on the ACME co.'s network. In particular:

1. A VPN for the road warriors employees;
2. A VPN tunnel between the Main and the Internal routers.

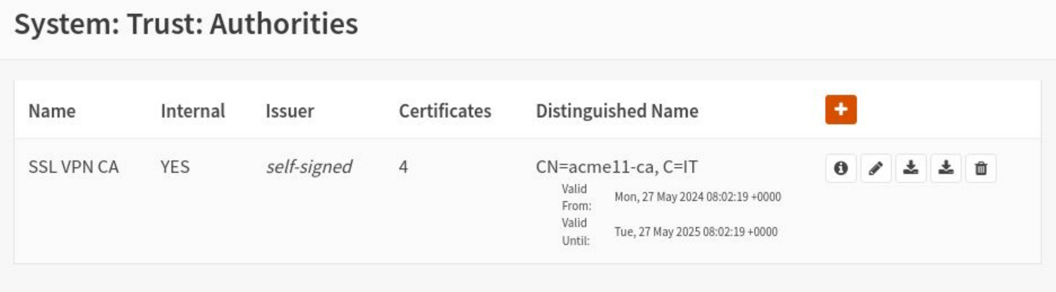
Regarding the first service, the requirements specify that the VPN has to be implemented with any of the possible options given by opnsense (OpenVPN, Wireguard or IPSec). We opted for implementing it using **OpenVPN** because we studied it during the course and felt more comfortable in using it. The second service instead must be implemented with **IPSec**.

2 VPN setup for the road warriors

In order to setup the service we followed the OPNsense official documentation for the setup of a SSL VPN Road Warrior.

2.1 Add Certificate Authority

The VPN server (that we are going to create later) needs a Certificate Authority to sign client or server certificates. In order to setup a new Certificate Authority we need to go to System → Trust → Authorities and click Add in the top right corner of the form. We named it "SSL VPN CA" and selected the method "Create an internal Certificate Authority".

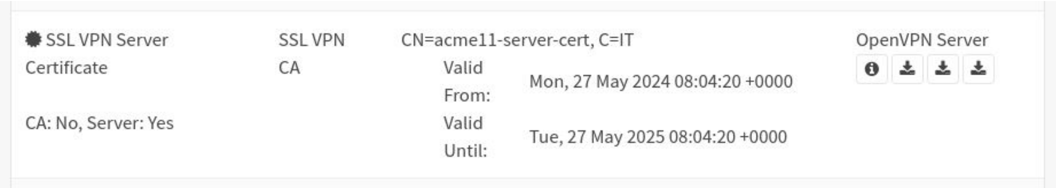


System: Trust: Authorities

Name	Internal	Issuer	Certificates	Distinguished Name	+
SSL VPN CA	YES	*self-signed*	4	CN=acme11-ca, C=IT Valid From: Mon, 27 May 2024 08:02:19 +0000 Valid Until: Tue, 27 May 2025 08:02:19 +0000	

2.2 Create a Server Certificate

After creating the Certificate Authority we also need a certificate. To create it, we have to go to System → Trust → Certificates and click Add in the upper right corner of the form. We named it "SSL VPN Server Certificate".



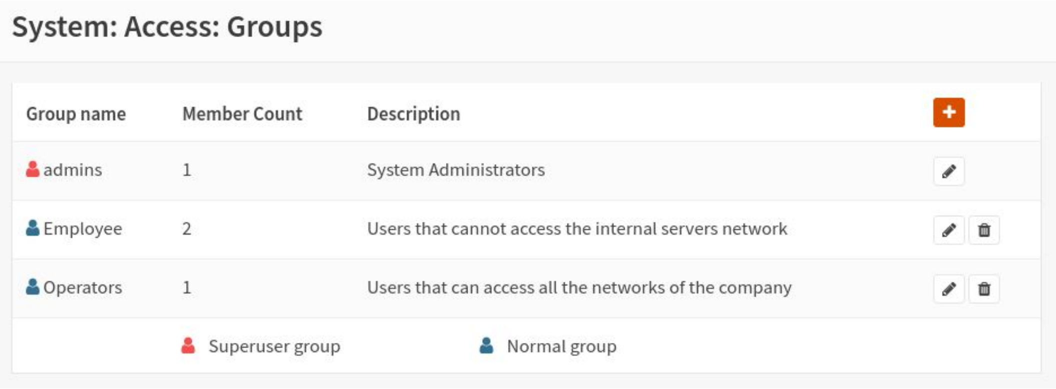
SSL VPN Server Certificate CA: No, Server: Yes	SSL VPN CA	CN=acme11-server-cert, C=IT Valid From: Mon, 27 May 2024 08:04:20 +0000 Valid Until: Tue, 27 May 2025 08:04:20 +0000	OpenVPN Server

2.3 Add the Users

The requirements of the assignment specify that the ACME co.'s network has two types of road warrior users: **operators** and **employees**. The operators can access all the networks of the company, while the empolyees can not access the internal server network. We have to create the following users:

1. Alice, operator
2. Bob, employee
3. Charles, employee

To add a new user we need to go to System → Access → Users and click Add in the top right corner. While inserting the information about the user we need to check the option "Click to create a user certificate" so that, after clicking on Save, we will be redirected to create the User Certificate by filling in the Certificate form. In order to assign the appropriate roles to our users (Alice is an operator while Bob and Charles are employee) we need to define two groups. To create a group we have to go to System → Access → Groups where, after defining a new group, we can also assign specific users to the group that we are currently creating.



System: Access: Groups

Group name	Member Count	Description
admins	1	System Administrators
Employee	2	Users that cannot access the internal servers network
Operators	1	Users that can access all the networks of the company

Superuser group   Normal group

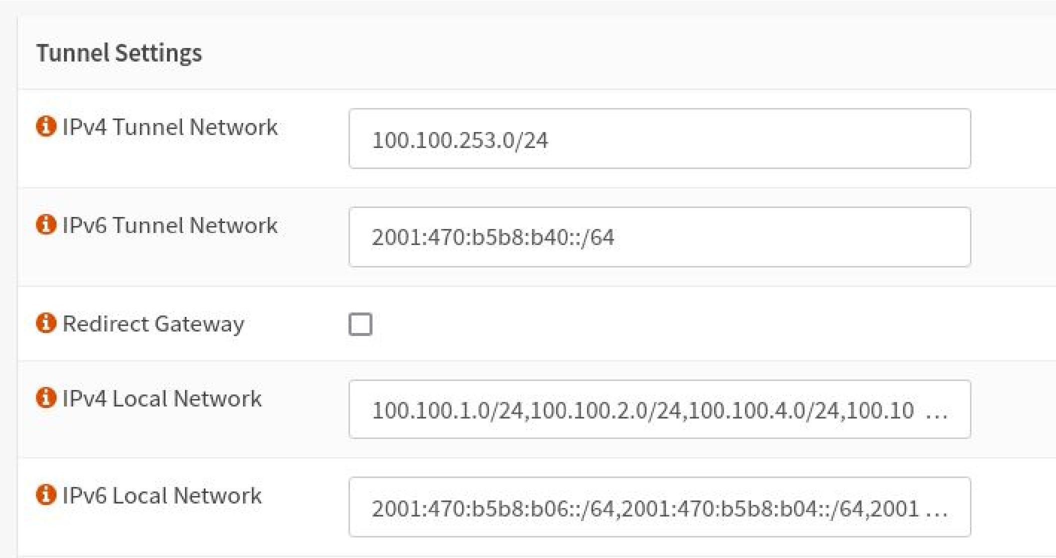
2.4 Add SSL Server

At this point we can create a VPN server on the Main firewall. This can be done by accessing OPNsense, on the main firewall, and then going to VPN → OpenVPN → Servers and click Add in the top right corner of the form.

We setup the server with the following configurations:

- Description: RoadWarrior VPN
- Server Mode: Remote Access (SSL/TLS + User Auth)
- Backend for authentication: Local Database
- Protocol: UDP (the default one for OpenVPN)
- Interface: WAN, since the road warrior are outside the ACME network all their traffic will have to go through the WAN interface
- Local port: 1194 (openVPN UDP port)
- TLS Authentication: Enabled - Authentication only
- Peer Certificate Authority: SSL VPN CA (the CA we created before)
- Server Certificate: SSL VPN Server Certificate (previously created)

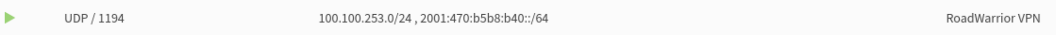
Regarding the **Tunnel Settings** we did the following:



Tunnel Settings

IPv4 Tunnel Network	100.100.253.0/24
IPv6 Tunnel Network	2001:470:b5b8:b40::/64
Redirect Gateway	□
IPv4 Local Network	100.100.1.0/24,100.100.2.0/24,100.100.4.0/24,100.10 …
IPv6 Local Network	2001:470:b5b8:b06::/64,2001:470:b5b8:b04::/64,2001 …

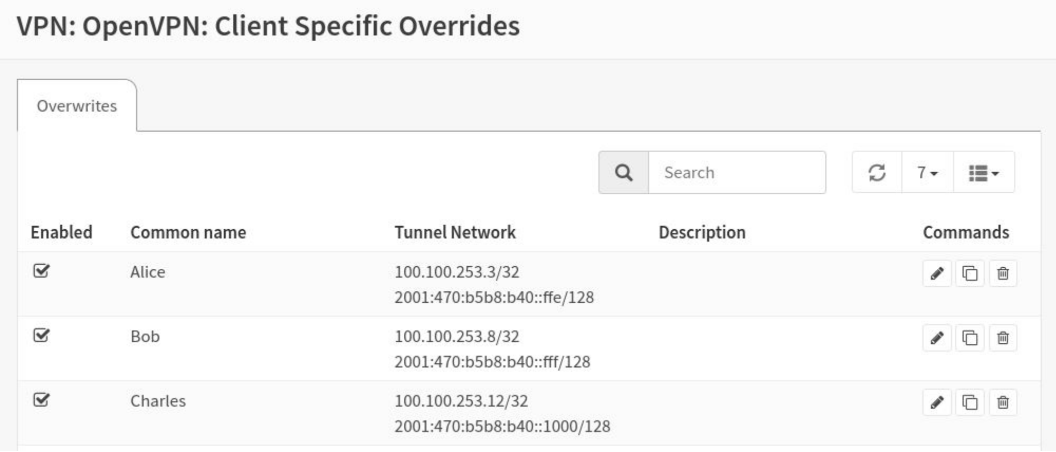
The **IPv4 Tunnel Network** (100.100.253.0/24) is the virtual subnet that the VPN will create. The **IPv4 Local Network** is the list of all the ACME subnets that will be reachable from the new VPN subnet. We did the same for IPv6. Finally, clicking on Save the new server will be added. This is the final result:



▶	UDP / 1194	100.100.253.0/24 , 2001:470:b5b8:b40::/64	RoadWarrior VPN

At this point we need to restrict the access of the users (previously created) to the ACME network. In particular, the operators can access all the networks of the company, while the empolyees can not access the internal server network. The vpn server assigns the IP addresses of his hosts dynamically choosing the IPs available in the VPN subnet pool. This makes the task of limiting their access inside the ACME network more difficult.

Therefore we decided to assign to each client a specific static IP address. To do so, from OPNsense we went to VPN → OpenVPN → Client Specific Overrides and we added three rules (one for each road warrior) by specifying the Server ”RoadWarrior VPN (1194/UDP), the name of the road warrior (respectively Alice, Bob and Charles), the IPv4 and IPv6 Tunnel Network. This is the result:



VPN: OpenVPN: Client Specific Overrides

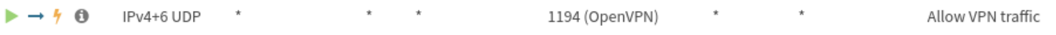
Enabled	Common name	Tunnel Network	Description	Commands
☑	Alice	100.100.253.3/32 2001:470:b5b8:b40::ffe/128		
☑	Bob	100.100.253.8/32 2001:470:b5b8:b40::fff/128		
☑	Charles	100.100.253.12/32 2001:470:b5b8:b40::1000/128		

So, we assigned to Alice 100.100.253.3/32 (since she’s an operator), to Bob 100.100.253.8/32 and to Charles 100.100.253.12/32 (since they are employees).

This allows us to give each role the necessary permissions by creating aliases inside the main firewall. We went to Firewall → Aliases and created the aliases “Operators”, with content the IPv4 address associated with Alice, and “Employees”, with content the IPv4 addresses associated with Bob and Charles.

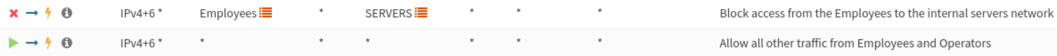
2.5 Firewall Rules

In order to allow SSL VPN client connections, we should allow access to the OpenVPN server port on the WAN interface of the Main Firewall. For our configuration we only use one server, accessible on UDP port 1194. So, from OPNsense on the main firewall we need to go to Firewall → Rules → WAN and define the following rule:



▶ → ϟ ⓘ	IPv4+6 UDP	*	*	*	1194 (OpenVPN)	*	*	Allow VPN traffic

The assignment specifies that “the empolyees can not access the internal server network” so we need to block the access to them. To accomplish this result, we defined a specific deny rule, using the alias “Employees”, in Firewall → Rules → OpenVPN. This rule blocks all the traffic originating from the employees, identified by their static IP addresses defined in the alias, and directed to the SERVERS subnet. Then, we can simply add a rule to allow all the IPv4 and IPv6 traffic.

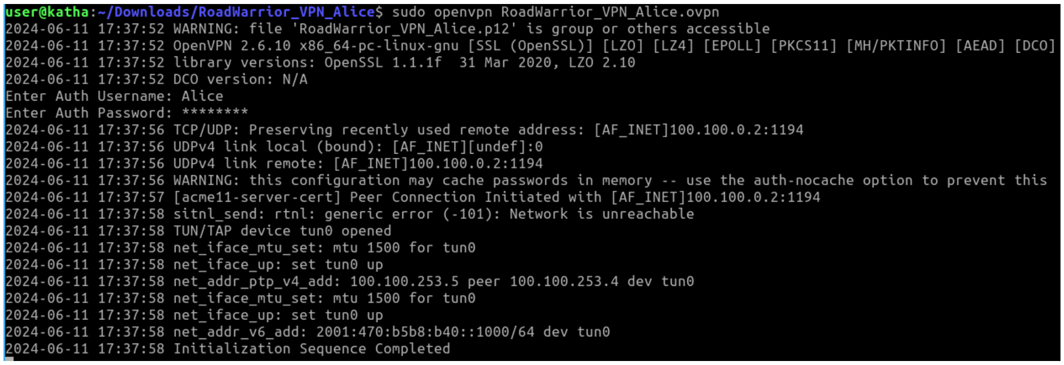


✕ → ϟ ⓘ	IPv4+6 *	Employees ☰	*	SERVERS ☰	*	*	*	Block access from the Employees to the internal servers network
▶ → ϟ ⓘ	IPv4+6 *	*	*	*	*	*	*	Allow all other traffic from Employees and Operators

2.6 Testing

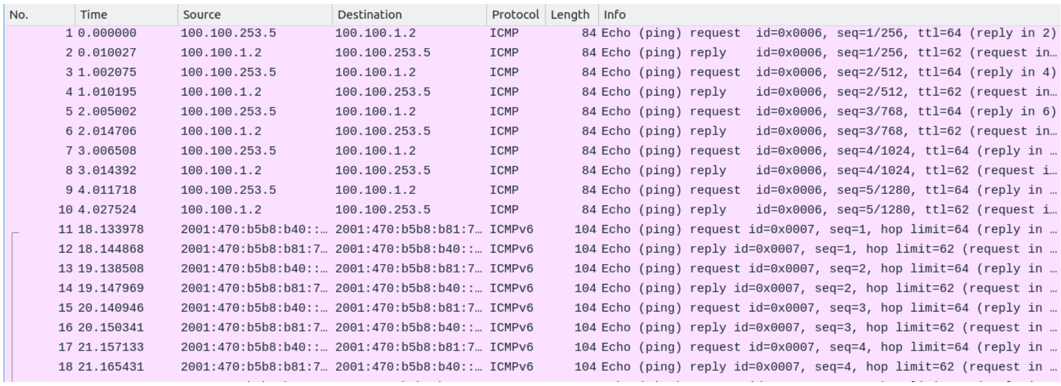
In order to test that the VPN is working as expected, we need to download the certificate of a user (Alice, Bob or Charles) that allows us to connect to the VPN as that host. To download the certificates we need to go to VPN → OpenVPN → Client Export.

We can connect to the VPN tunnel as Alice by using the command `sudo openvpn Roadwarriors_VPN_Alice.ovpn` from our virtual machine, and we can see that everything is working correctly.



```
user@katha:~/Downloads/RoadWarrior_VPN_Alice$ sudo openvpn RoadWarrior_VPN_Alice.ovpn
2024-06-11 17:37:52 WARNING: file 'RoadWarrior_VPN_Alice.p12' is group or others accessible
2024-06-11 17:37:52 OpenVPN 2.6.10 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] [DCO]
2024-06-11 17:37:52 library versions: OpenSSL 1.1.1f  31 Mar 2020, LZO 2.10
2024-06-11 17:37:52 DCO version: N/A
Enter Auth Username: Alice
Enter Auth Password: ********
2024-06-11 17:37:56 TCP/UDP: Preserving recently used remote address: [AF_INET]100.100.0.2:1194
2024-06-11 17:37:56 UDPv4 link local (bound): [AF_INET][undef]:0
2024-06-11 17:37:56 UDPv4 link remote: [AF_INET]100.100.0.2:1194
2024-06-11 17:37:56 WARNING: this configuration may cache passwords in memory -- use the auth-nocache option to prevent this
2024-06-11 17:37:57 [acme11-server-cert] Peer Connection Initiated with [AF_INET]100.100.0.2:1194
2024-06-11 17:37:58 sitnl_send: rtnl: generic error (-101): Network is unreachable
2024-06-11 17:37:58 TUN/TAP device tun0 opened
2024-06-11 17:37:58 net_iface_mtu_set: mtu 1500 for tun0
2024-06-11 17:37:58 net_iface_up: set tun0 up
2024-06-11 17:37:58 net_addr_ptp_v4_add: 100.100.253.5 peer 100.100.253.4 dev tun0
2024-06-11 17:37:58 net_iface_mtu_set: mtu 1500 for tun0
2024-06-11 17:37:58 net_iface_up: set tun0 up
2024-06-11 17:37:58 net_addr_v6_add: 2001:470:b5b8:b40::1000/64 dev tun0
2024-06-11 17:37:58 Initialization Sequence Completed
```

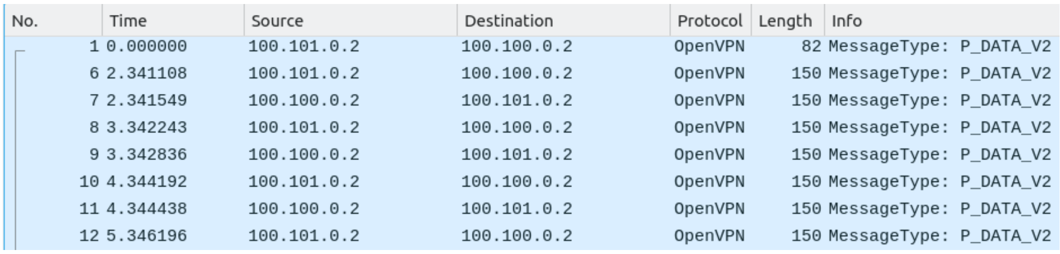
We can now ping the DNS server from our `tun0` interface, using both IPv4 and IPv6 addresses:



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	100.100.253.5	100.100.1.2	ICMP	84	Echo (ping) request id=0x0006, seq=1/256, ttl=64 (reply in 2)
2	0.010027	100.100.1.2	100.100.253.5	ICMP	84	Echo (ping) reply id=0x0006, seq=1/256, ttl=62 (request in…
3	1.002075	100.100.253.5	100.100.1.2	ICMP	84	Echo (ping) request id=0x0006, seq=2/512, ttl=64 (reply in 4)
4	1.010195	100.100.1.2	100.100.253.5	ICMP	84	Echo (ping) reply id=0x0006, seq=2/512, ttl=62 (request in…
5	2.005002	100.100.253.5	100.100.1.2	ICMP	84	Echo (ping) request id=0x0006, seq=3/768, ttl=64 (reply in 6)
6	2.014706	100.100.1.2	100.100.253.5	ICMP	84	Echo (ping) reply id=0x0006, seq=3/768, ttl=62 (request in…
7	3.006508	100.100.253.5	100.100.1.2	ICMP	84	Echo (ping) request id=0x0006, seq=4/1024, ttl=64 (reply in …
8	3.014392	100.100.1.2	100.100.253.5	ICMP	84	Echo (ping) reply id=0x0006, seq=4/1024, ttl=62 (request i…
9	4.011718	100.100.253.5	100.100.1.2	ICMP	84	Echo (ping) request id=0x0006, seq=5/1280, ttl=64 (reply in …
10	4.027524	100.100.1.2	100.100.253.5	ICMP	84	Echo (ping) reply id=0x0006, seq=5/1280, ttl=62 (request i…
11	18.133978	2001:470:b5b8:b40::…	2001:470:b5b8:b81:7…	ICMPv6	104	Echo (ping) request id=0x0007, seq=1, hop limit=64 (reply in …
12	18.144868	2001:470:b5b8:b81:7…	2001:470:b5b8:b40::…	ICMPv6	104	Echo (ping) reply id=0x0007, seq=1, hop limit=62 (request in …
13	19.138508	2001:470:b5b8:b40::…	2001:470:b5b8:b81:7…	ICMPv6	104	Echo (ping) request id=0x0007, seq=2, hop limit=64 (reply in …
14	19.147969	2001:470:b5b8:b81:7…	2001:470:b5b8:b40::…	ICMPv6	104	Echo (ping) reply id=0x0007, seq=2, hop limit=62 (request in …
15	20.140946	2001:470:b5b8:b40::…	2001:470:b5b8:b81:7…	ICMPv6	104	Echo (ping) request id=0x0007, seq=3, hop limit=64 (reply in …
16	20.150341	2001:470:b5b8:b81:7…	2001:470:b5b8:b40::…	ICMPv6	104	Echo (ping) reply id=0x0007, seq=3, hop limit=62 (request in …
17	21.157133	2001:470:b5b8:b40::…	2001:470:b5b8:b81:7…	ICMPv6	104	Echo (ping) request id=0x0007, seq=4, hop limit=64 (reply in …
18	21.165431	2001:470:b5b8:b81:7…	2001:470:b5b8:b40::…	ICMPv6	104	Echo (ping) reply id=0x0007, seq=4, hop limit=62 (request in …

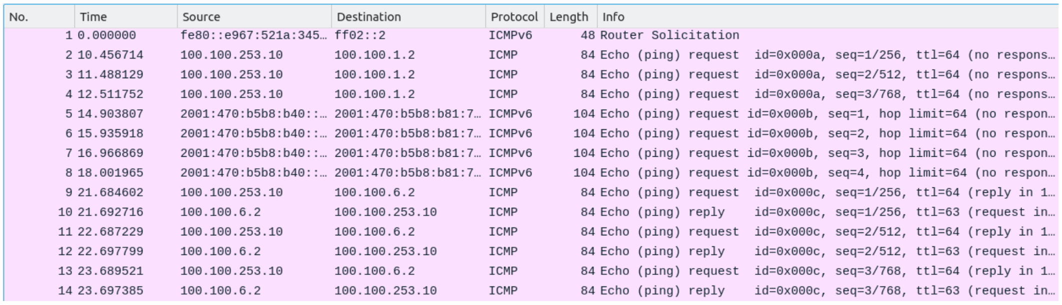
As expected, the packets are displayed in cleartext, since we are capturing inside the OpenVPN tunnel.

We can now capture the packets on the `tap0` interface and see that they are correctly encapsulated and encrypted by the VPN.



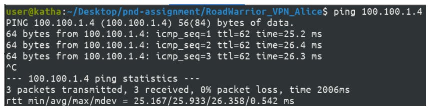
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	100.101.0.2	100.100.0.2	OpenVPN	82	MessageType: P_DATA_V2
6	2.341108	100.101.0.2	100.100.0.2	OpenVPN	150	MessageType: P_DATA_V2
7	2.341549	100.100.0.2	100.101.0.2	OpenVPN	150	MessageType: P_DATA_V2
8	3.342243	100.101.0.2	100.100.0.2	OpenVPN	150	MessageType: P_DATA_V2
9	3.342836	100.100.0.2	100.101.0.2	OpenVPN	150	MessageType: P_DATA_V2
10	4.344192	100.101.0.2	100.100.0.2	OpenVPN	150	MessageType: P_DATA_V2
11	4.344438	100.100.0.2	100.101.0.2	OpenVPN	150	MessageType: P_DATA_V2
12	5.346196	100.101.0.2	100.100.0.2	OpenVPN	150	MessageType: P_DATA_V2

To test access control for operators and employees, we have seen that Alice is able to ping the DNS server inside the SERVES subnet. We can try to do the same thing with Bob (employee) and we can verify that we do not get the replys to our echo requests, but we can correctly ping the webserver:



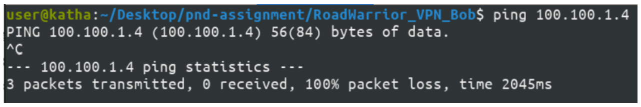
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	fe80::e967:521a:345…	ff02::2	ICMPv6	48	Router Solicitation
2	10.456714	100.100.253.10	100.100.1.2	ICMP	84	Echo (ping) request id=0x000a, seq=1/256, ttl=64 (no respons…
3	11.488129	100.100.253.10	100.100.1.2	ICMP	84	Echo (ping) request id=0x000a, seq=2/512, ttl=64 (no respons…
4	12.511752	100.100.253.10	100.100.1.2	ICMP	84	Echo (ping) request id=0x000a, seq=3/768, ttl=64 (no respons…
5	14.903807	2001:470:b5b8:b40::…	2001:470:b5b8:b81:7…	ICMPv6	104	Echo (ping) request id=0x000b, seq=1, hop limit=64 (no respon…
6	15.935918	2001:470:b5b8:b40::…	2001:470:b5b8:b81:7…	ICMPv6	104	Echo (ping) request id=0x000b, seq=2, hop limit=64 (no respon…
7	16.966869	2001:470:b5b8:b40::…	2001:470:b5b8:b81:7…	ICMPv6	104	Echo (ping) request id=0x000b, seq=3, hop limit=64 (no respon…
8	18.001965	2001:470:b5b8:b40::…	2001:470:b5b8:b81:7…	ICMPv6	104	Echo (ping) request id=0x000b, seq=4, hop limit=64 (no respon…
9	21.684602	100.100.253.10	100.100.6.2	ICMP	84	Echo (ping) request id=0x000c, seq=1/256, ttl=64 (reply in 1…
10	21.692716	100.100.6.2	100.100.253.10	ICMP	84	Echo (ping) reply id=0x000c, seq=1/256, ttl=63 (request in…
11	22.687229	100.100.253.10	100.100.6.2	ICMP	84	Echo (ping) request id=0x000c, seq=2/512, ttl=64 (reply in 1…
12	22.697799	100.100.6.2	100.100.253.10	ICMP	84	Echo (ping) reply id=0x000c, seq=2/512, ttl=63 (request in…
13	23.689521	100.100.253.10	100.100.6.2	ICMP	84	Echo (ping) request id=0x000c, seq=3/768, ttl=64 (reply in 1…
14	23.697385	100.100.6.2	100.100.253.10	ICMP	84	Echo (ping) reply id=0x000c, seq=3/768, ttl=63 (request in…

Another test was performed in the following screenshots, where we can see that Alice is able to ping a the Greenbone server, whereas Bob cannot do that.



```
user@katha:~/Desktop/pnd-assignment/RoadWarrior_VPN_Alice$ ping 100.100.1.4
PING 100.100.1.4 (100.100.1.4) 56(84) bytes of data.
64 bytes from 100.100.1.4: icmp_seq=1 ttl=62 time=25.2 ms
64 bytes from 100.100.1.4: icmp_seq=2 ttl=62 time=26.4 ms
64 bytes from 100.100.1.4: icmp_seq=3 ttl=62 time=26.3 ms
^C
--- 100.100.1.4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2006ms
rtt min/avg/max/mdev = 25.167/25.933/26.358/0.542 ms
```

Figure 1: Alice can ping Greenbone.



```
user@katha:~/Desktop/pnd-assignment/RoadWarrior_VPN_Bob$ ping 100.100.1.4
PING 100.100.1.4 (100.100.1.4) 56(84) bytes of data.
^C
--- 100.100.1.4 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2045ms
```

Figure 2: Bob can’t ping Greenbone.

3 VPN for the Main-Internal VPN tunnel

Regarding the second service, the assignment requires to setup a VPN tunnel between the Main and the Internal routers, since the actual connection is running on an insecure medium: any packet running between the two routers should go through an IPsec tunnel.

To configure a VPN tunnel between the two routers (a site-to-site VPN) using IPsec, we followed the official documentation provided by OPNsense. In order to implement an IPsec tunnel between the routers we need to configure both the Main and the Internal Firewalls.

3.1 Main Firewall Router configuration

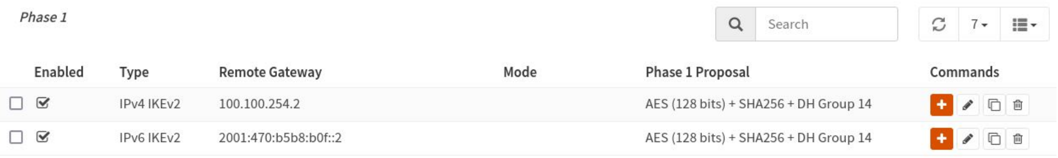
We need to go to VPN → IPsec → Tunnel Settings

- Phase 1

  We followed these steps:

  - Selected the “INTERNAL” interface
  - Set the remote gateway as 100.100.254.2 which is the IP of the “EXTERNAL” interface of the Internal Firewall
  - Selected Mutual PSK as Authentication Method
  - Inserted a Pre-Shared Key (that we generated online)
  - Selected an Encryption algorithm

  We did the same also for IPv6. This is the final result:



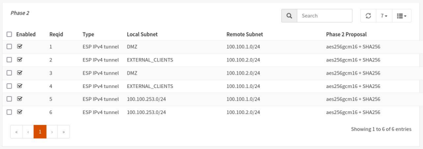
Phase 1

Search | 7 ▾

	Enabled	Type	Remote Gateway	Mode	Phase 1 Proposal	Commands
☐	☑	IPv4 IKEv2	100.100.254.2		AES (128 bits) + SHA256 + DH Group 14	+ ✎ ⧉ 🗑
☐	☑	IPv6 IKEv2	2001:470:b5b8:b0f::2		AES (128 bits) + SHA256 + DH Group 14	+ ✎ ⧉ 🗑

- Phase 2 (done for every pair of local and remote subnets)
  - Set ESP protocol for the secure association
  - Tick the box “Enable IPsec”

  This is the result for both IPv4 and IPv6:



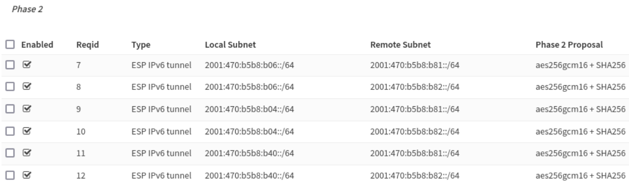
Phase 2

Search | 7 ▾

☐	Enabled	Reqid	Type	Local Subnet	Remote Subnet	Phase 2 Proposal
☐	☑	1	ESP IPv4 tunnel	DMZ	100.100.1.0/24	aes256gcm16 + SHA256
☐	☑	2	ESP IPv4 tunnel	EXTERNAL_CLIENTS	100.100.2.0/24	aes256gcm16 + SHA256
☐	☑	3	ESP IPv4 tunnel	DMZ	100.100.2.0/24	aes256gcm16 + SHA256
☐	☑	4	ESP IPv4 tunnel	EXTERNAL_CLIENTS	100.100.1.0/24	aes256gcm16 + SHA256
☐	☑	5	ESP IPv4 tunnel	100.100.253.0/24	100.100.1.0/24	aes256gcm16 + SHA256
☐	☑	6	ESP IPv4 tunnel	100.100.253.0/24	100.100.2.0/24	aes256gcm16 + SHA256

« ‹ 1 › »

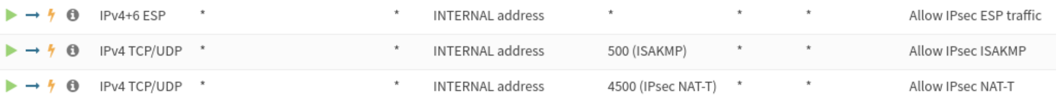
Showing 1 to 6 of 6 entries



Phase 2

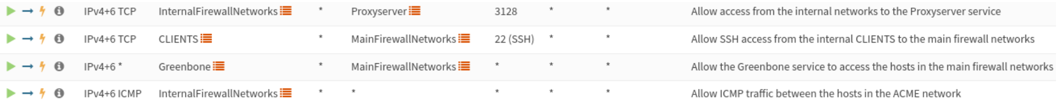
☐	Enabled	Reqid	Type	Local Subnet	Remote Subnet	Phase 2 Proposal
☐	☑	7	ESP IPv6 tunnel	2001:470:b5b8:b06::/64	2001:470:b5b8:b81::/64	aes256gcm16 + SHA256
☐	☑	8	ESP IPv6 tunnel	2001:470:b5b8:b06::/64	2001:470:b5b8:b82::/64	aes256gcm16 + SHA256
☐	☑	9	ESP IPv6 tunnel	2001:470:b5b8:b04::/64	2001:470:b5b8:b81::/64	aes256gcm16 + SHA256
☐	☑	10	ESP IPv6 tunnel	2001:470:b5b8:b04::/64	2001:470:b5b8:b82::/64	aes256gcm16 + SHA256
☐	☑	11	ESP IPv6 tunnel	2001:470:b5b8:b40::/64	2001:470:b5b8:b81::/64	aes256gcm16 + SHA256
☐	☑	12	ESP IPv6 tunnel	2001:470:b5b8:b40::/64	2001:470:b5b8:b82::/64	aes256gcm16 + SHA256

We also added the following rules to the INTERNAL interface of the fierwall, in order to allow the flow of the ESP traffic and the packets for the IKE protocol, as outlined in the OPNsense documentation.



▶ → ϟ ⓘ	IPv4+6 ESP	*	*	INTERNAL address	*	*	*	Allow IPsec ESP traffic
▶ → ϟ ⓘ	IPv4 TCP/UDP	*	*	INTERNAL address	500 (ISAKMP)	*	*	Allow IPsec ISAKMP
▶ → ϟ ⓘ	IPv4 TCP/UDP	*	*	INTERNAL address	4500 (IPsec NAT-T)	*	*	Allow IPsec NAT-T

Moreover, we moved the other rules of the INTERNAL interface to the IPsec interface of the firewall, because now the traffic between the two firewalls is going through the IPsec tunnel.



▶ → ϟ ⓘ	IPv4+6 TCP	InternalFirewallNetworks ☰	*	Proxyserver ☰	3128	*	*	Allow access from the internal networks to the Proxyserver service
▶ → ϟ ⓘ	IPv4+6 TCP	CLIENTS ☰	*	MainFirewallNetworks ☰	22 (SSH)	*	*	Allow SSH access from the internal CLIENTS to the main firewall networks
▶ → ϟ ⓘ	IPv4+6 *	Greenbone ☰	*	MainFirewallNetworks ☰	*	*	*	Allow the Greenbone service to access the hosts in the main firewall networks
▶ → ϟ ⓘ	IPv4+6 ICMP	InternalFirewallNetworks ☰	*	*	*	*	*	Allow ICMP traffic between the hosts in the ACME network

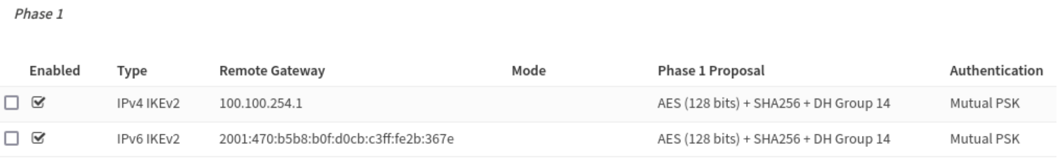
3.2 Internal Firewall Router configuration

We followed almost the same steps as before. We need to go to VPN → IPsec → Tunnel Settings, on the Internal Firewall.

- Phase 1
  - Selected the interface “EXTERNAL”
  - Set the remote gateway as 100.100.254.1 which is the IP of the “INTERNAL” interface of the Main Firewall

- Selected Mutual PSK as Authentication Method
- Inserted the same Pre-Shared Key as before (generated online)
- Selected the same Encryption algorithm as before

We did the same also for IPv6. This is the final result:

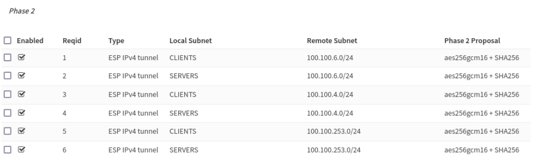


Phase 1

	Enabled	Type	Remote Gateway	Mode	Phase 1 Proposal	Authentication
☐	☑	IPv4 IKEv2	100.100.254.1		AES (128 bits) + SHA256 + DH Group 14	Mutual PSK
☐	☑	IPv6 IKEv2	2001:470:b5b8:b0f:d0cb:c3ff:fe2b:367e		AES (128 bits) + SHA256 + DH Group 14	Mutual PSK

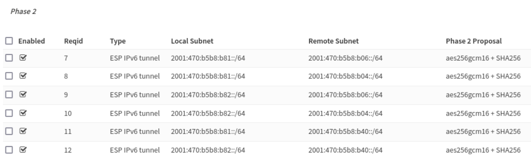
- Phase 2 (done for every pair of local and remote subnets)
  - Set ESP protocol for the secure association
  - Tick the box "Enable IPsec"

This is the result for both IPv4 and IPv6:



Phase 2

☐	Enabled	Reqid	Type	Local Subnet	Remote Subnet	Phase 2 Proposal
☐	☑	1	ESP IPv4 tunnel	CLIENTS	100.100.6.0/24	aes256gcm16 + SHA256
☐	☑	2	ESP IPv4 tunnel	SERVERS	100.100.6.0/24	aes256gcm16 + SHA256
☐	☑	3	ESP IPv4 tunnel	CLIENTS	100.100.4.0/24	aes256gcm16 + SHA256
☐	☑	4	ESP IPv4 tunnel	SERVERS	100.100.4.0/24	aes256gcm16 + SHA256
☐	☑	5	ESP IPv4 tunnel	CLIENTS	100.100.253.0/24	aes256gcm16 + SHA256
☐	☑	6	ESP IPv4 tunnel	SERVERS	100.100.253.0/24	aes256gcm16 + SHA256



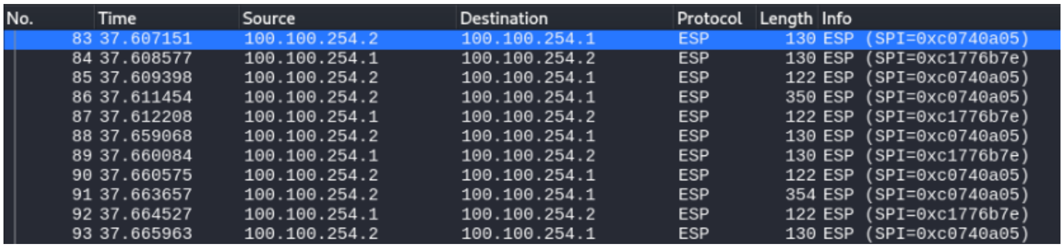
Phase 2

☐	Enabled	Reqid	Type	Local Subnet	Remote Subnet	Phase 2 Proposal
☐	☑	7	ESP IPv6 tunnel	2001:470:b5b8:b81::/64	2001:470:b5b8:b06::/64	aes256gcm16 + SHA256
☐	☑	8	ESP IPv6 tunnel	2001:470:b5b8:b81::/64	2001:470:b5b8:b04::/64	aes256gcm16 + SHA256
☐	☑	9	ESP IPv6 tunnel	2001:470:b5b8:b82::/64	2001:470:b5b8:b06::/64	aes256gcm16 + SHA256
☐	☑	10	ESP IPv6 tunnel	2001:470:b5b8:b82::/64	2001:470:b5b8:b04::/64	aes256gcm16 + SHA256
☐	☑	11	ESP IPv6 tunnel	2001:470:b5b8:b81::/64	2001:470:b5b8:b40::/64	aes256gcm16 + SHA256
☐	☑	12	ESP IPv6 tunnel	2001:470:b5b8:b82::/64	2001:470:b5b8:b40::/64	aes256gcm16 + SHA256

We also added the same rules as the ones in the main firewall to the EXTERNAL interface of the internal firewall, to allow the flow of the ESP traffic and the packets involved in the IKE protocol, and we moved the other rules of this interface into the IPSec interface.

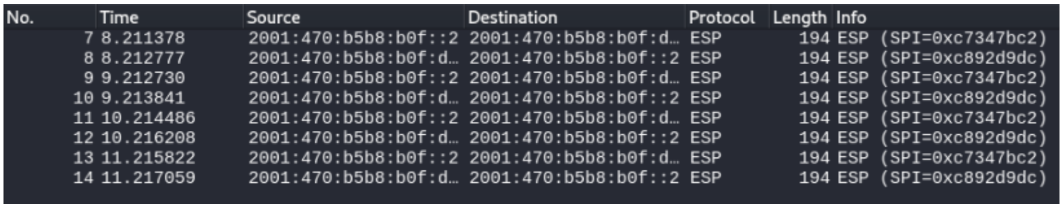
3.3 Testing

To test whether IPSec is working properly we can access a host from the web browser, e.g. from kali we access the webserver. The traffic generated is captured using `tcpdump` on the internal firewall, then using `wget` we transfer the `pcap` file to kali to analyze it with Wireshark. The result is displayed below, we can see that all the HTTP traffic is encapsulated into encrypted ESP packets.



```
No.   Time       Source          Destination     Protocol Length Info
   83 37.607151  100.100.254.2   100.100.254.1   ESP         130 ESP (SPI=0xc0740a05)
   84 37.608577  100.100.254.1   100.100.254.2   ESP         130 ESP (SPI=0xc1776b7e)
   85 37.609398  100.100.254.2   100.100.254.1   ESP         122 ESP (SPI=0xc0740a05)
   86 37.611454  100.100.254.2   100.100.254.1   ESP         350 ESP (SPI=0xc0740a05)
   87 37.612208  100.100.254.1   100.100.254.2   ESP         122 ESP (SPI=0xc1776b7e)
   88 37.659068  100.100.254.2   100.100.254.1   ESP         130 ESP (SPI=0xc0740a05)
   89 37.660084  100.100.254.1   100.100.254.2   ESP         130 ESP (SPI=0xc1776b7e)
   90 37.660575  100.100.254.2   100.100.254.1   ESP         122 ESP (SPI=0xc0740a05)
   91 37.663657  100.100.254.2   100.100.254.1   ESP         354 ESP (SPI=0xc0740a05)
   92 37.664527  100.100.254.1   100.100.254.2   ESP         122 ESP (SPI=0xc1776b7e)
   93 37.665963  100.100.254.2   100.100.254.1   ESP         130 ESP (SPI=0xc0740a05)
```

We can also verify that the IPSec tunnel is correctly working for IPv6 too, by capturing the traffic generated by pinging the webserver from kali and analyzing it with Wireshark (with the same procedure used before). The processed data is shown below, we can see that all the traffic is correctly encrypted and encapsulated using the ESP protocol.



```
No.   Time        Source                 Destination            Protocol  Length Info
    7 8.211378    2001:470:b5b8:b0f::2   2001:470:b5b8:b0f:d…   ESP          194 ESP (SPI=0xc7347bc2)
    8 8.212777    2001:470:b5b8:b0f:d…   2001:470:b5b8:b0f::2   ESP          194 ESP (SPI=0xc892d9dc)
    9 9.212730    2001:470:b5b8:b0f::2   2001:470:b5b8:b0f:d…   ESP          194 ESP (SPI=0xc7347bc2)
   10 9.213841    2001:470:b5b8:b0f:d…   2001:470:b5b8:b0f::2   ESP          194 ESP (SPI=0xc892d9dc)
   11 10.214486   2001:470:b5b8:b0f::2   2001:470:b5b8:b0f:d…   ESP          194 ESP (SPI=0xc7347bc2)
   12 10.216208   2001:470:b5b8:b0f:d…   2001:470:b5b8:b0f::2   ESP          194 ESP (SPI=0xc892d9dc)
   13 11.215822   2001:470:b5b8:b0f::2   2001:470:b5b8:b0f:d…   ESP          194 ESP (SPI=0xc7347bc2)
   14 11.217059   2001:470:b5b8:b0f:d…   2001:470:b5b8:b0f::2   ESP          194 ESP (SPI=0xc892d9dc)
```

4 Final Remarks

This assignment proved to be quite demanding, but we managed to complete every task. The challenges we faced allowed us to improve our problem-solving skills and deepen our understanding of the subjects seen during the lectures.