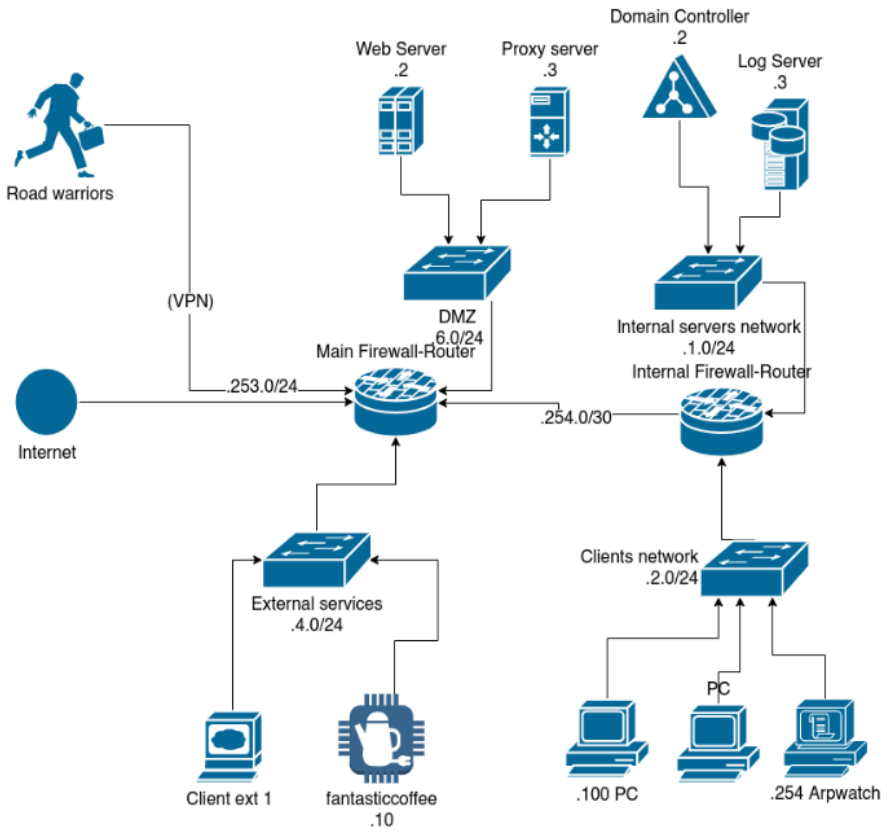
ACME_32_a1_report

Group number: 32
Students:

- Lilit Hasan: 1958055
- Matteo Piermartini: 1802597
- Aurora Polifemo: 1802485

Initial Brainstorming

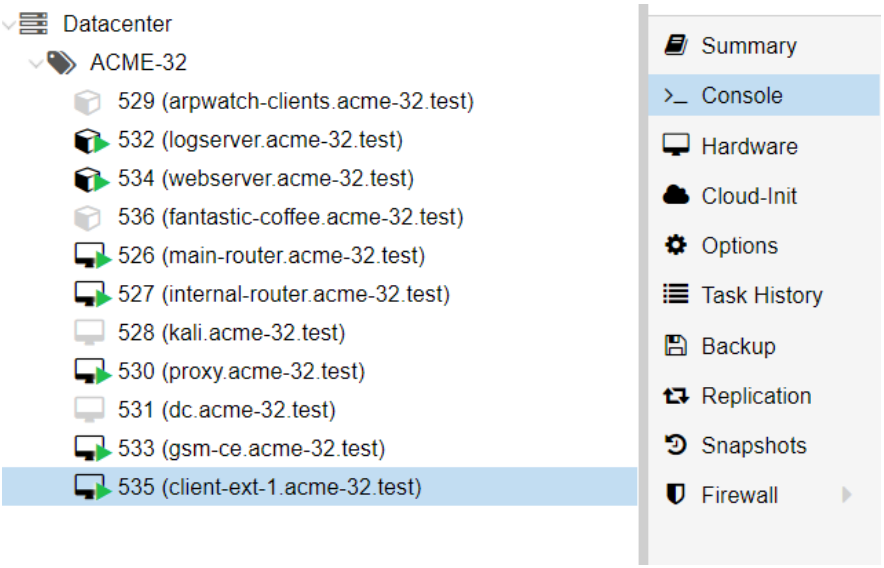


100.100.0.0/16 network

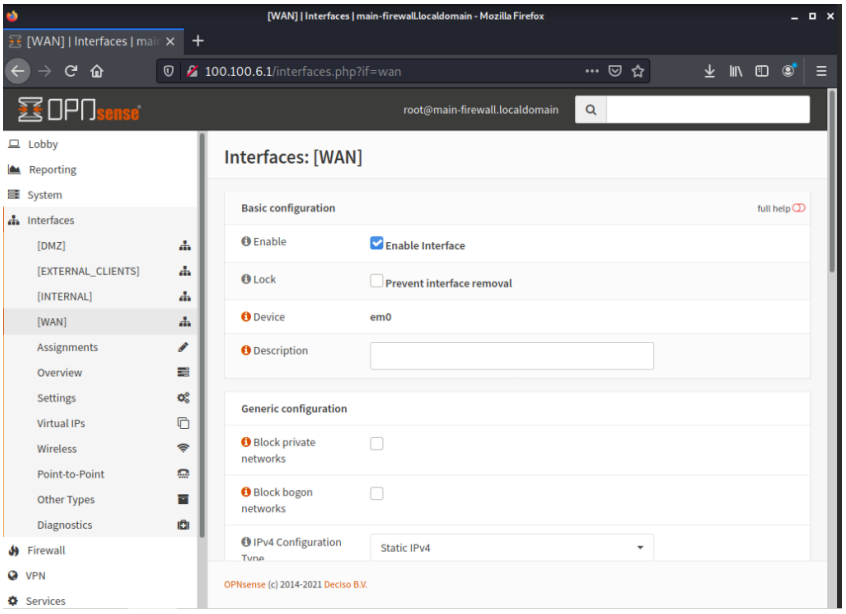
The first consideration we have to make is based on the network architecture. It is a network with two router/firewalls, the main one and the internal one. These two are connected through a point-to-point connection. Each one of them is connected to two other subnetworks, with a total of four subnetworks. The main router is connected to the Internet through a WAN too.
Our goals are to assign IPv6 addresses to each host, configure the DNS servers and manage the firewalls by applying them some security policies. For the first point we decided to use DHCPv6, for the second to use dnsmasq, while for the last one, we just use the rules of the firewall.

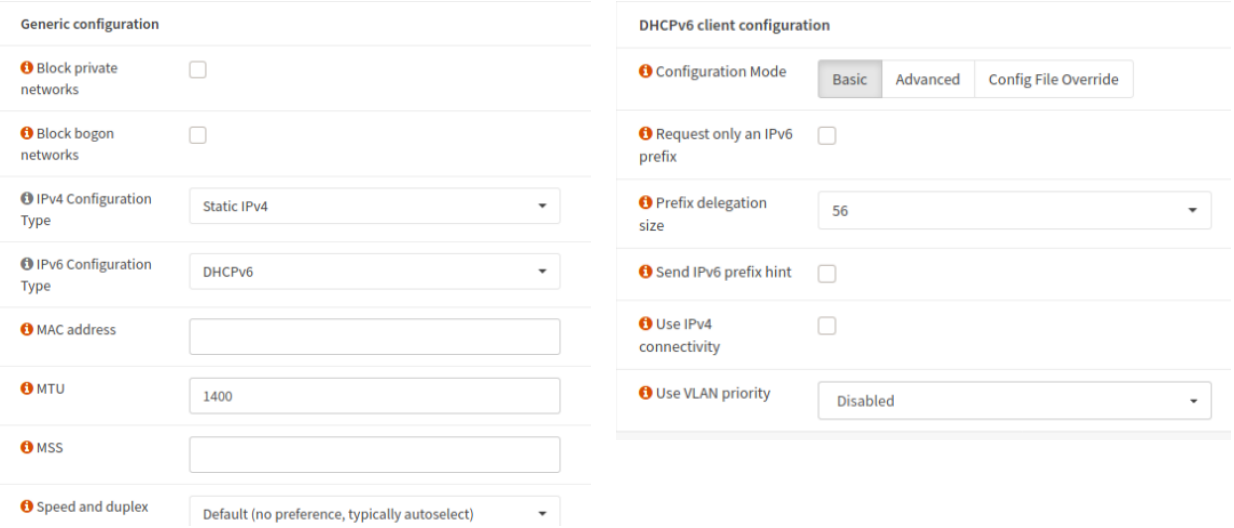
1. Set-up of the infrastructure for IPv6 addressing

The first step is to set IPv6 addressing to the hosts in the network, to do that we use *Prefix Delegation*, that is used to assign a network address prefix.
On the proxmox interface, we have to enable the Prefix Delegation from the Main Router to the ISP in the WAN, in order to allow the router to receive the prefix for a IPv6 network from the ISP.
To do that we have to select the *Virtual Machine 535 (client-ext-1.acme-32.test)* and on the *Console* we have to access the Main Router on the IP address 100.100.6.1.

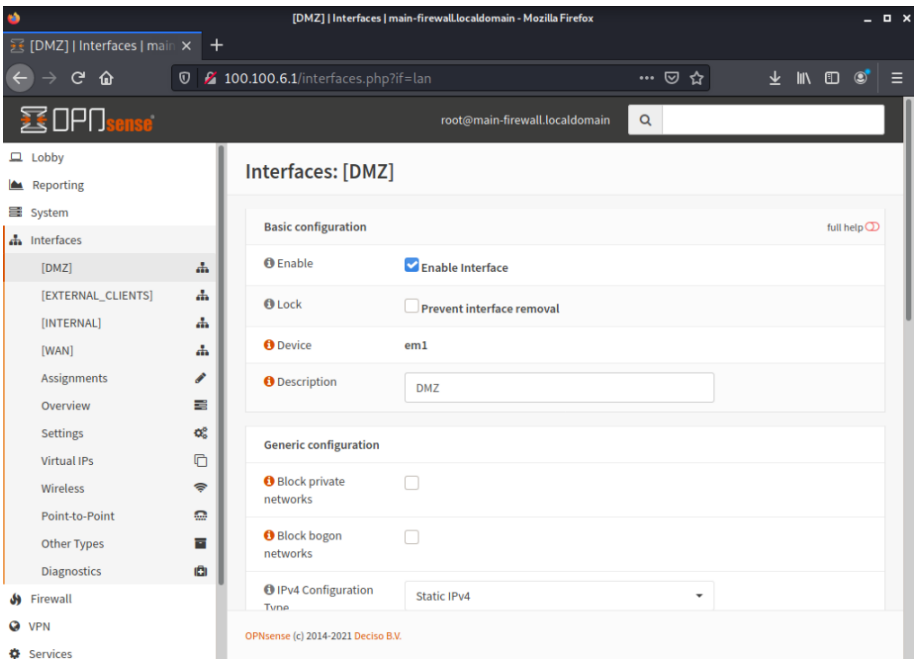


Then we have to go on the *Interfaces* tab and on *[WAN]*. On *Generic Configuration* we disable *Block Private Networks* and *Block Bogon Networks*, then we have to enable the *IPv6 Configuration Type* on *DHCPv6*. And on *DHCPv6 client configuration*, under *Prefix delegation size* we write 56, in order to get /56 prefixes.

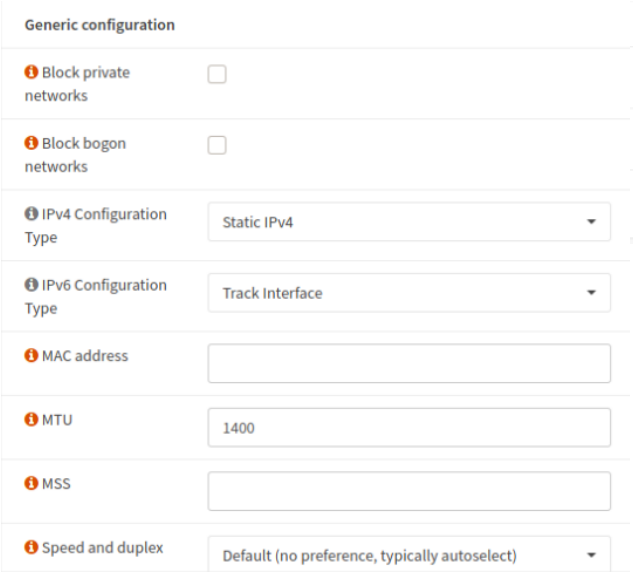


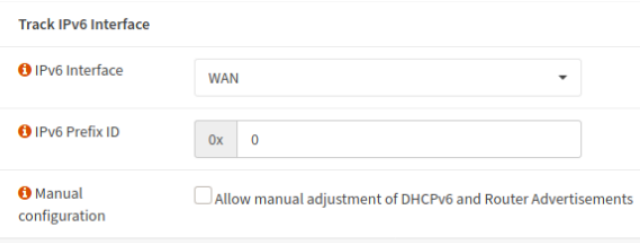


After the router receives the suggested IPv6 address prefix, it has to distribute them to the host in the DMZ. So, on the same host (*main router*) we choose *[DMZ]*.



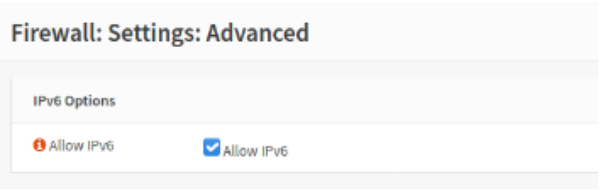
Under *Generic configuration*, we enable the *IPv6 Configuration Type* using *Track Interface* and then, in the *Track IPv6 Interface* under *IPv6 Interface* we put *WAN*.
Now the hosts in the DMZ should get the IPv6 addresses from the main router that get the prefixes from the ISP in the WAN. The *Web Server* got an IPv6 Address, while the *Proxy Server* didn't get any, so what we do is go to the *Virtual Machine 530* and check on the terminal the *disable_ipv6* flag in the "/proc/sys/net/ipv6/conf/all/" directory. It is set to 1, that means that it's disabled, so we write *echo > 0 /proc/sys/net/ipv6/conf/all/disable_ipv6* in order to set the flag to 0. We do the exact same thing for the dc. After that the *Proxy Server* and the *Domain Controller* have an IPv6.



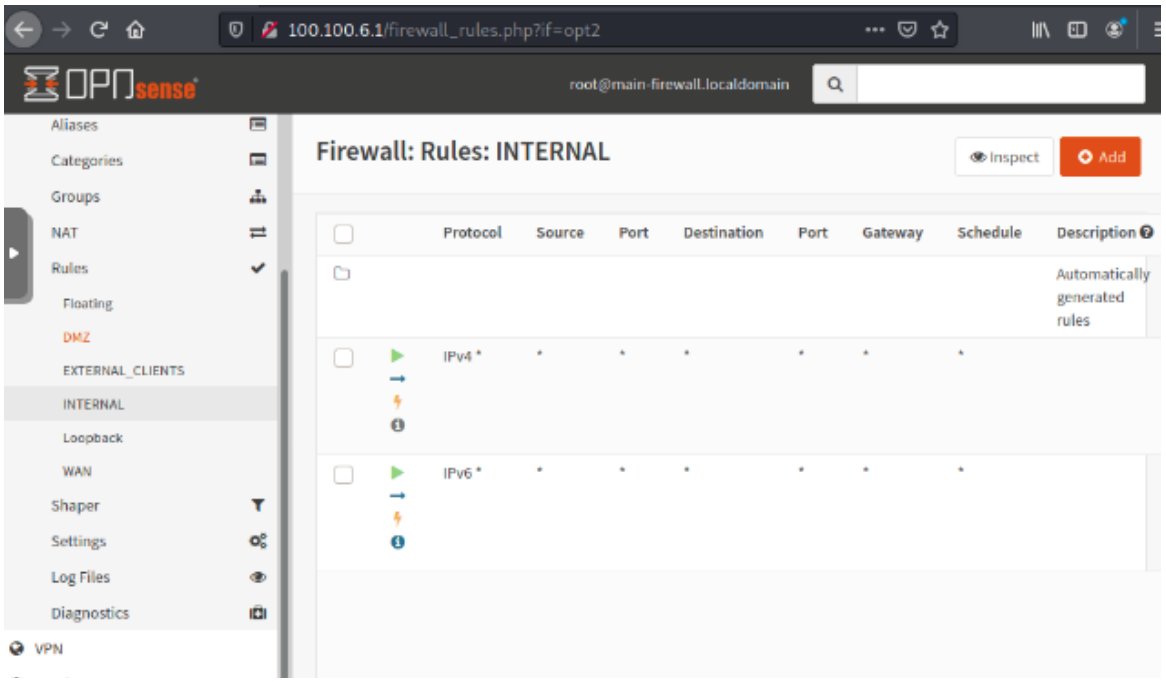


We need to assign an IPv6 address to all the remaining hosts in the subnetworks, to do so we make the same steps as before. In the *Main Router*, we set *IPv6 Configuration Type* to *Track Interface* in the *External* and *Internal* interfaces. We have to do the same in the *Internal Router*, we use *Track Interface* and set the *Prefix IDs* to 1 in the *Servers network* interface and 0 in the *Clients network* interface, while the *External* interface has the *IPv6 Configuration Type* on *DHCPv6* (instead of *Track Interface*) with *Prefix ID* /62.

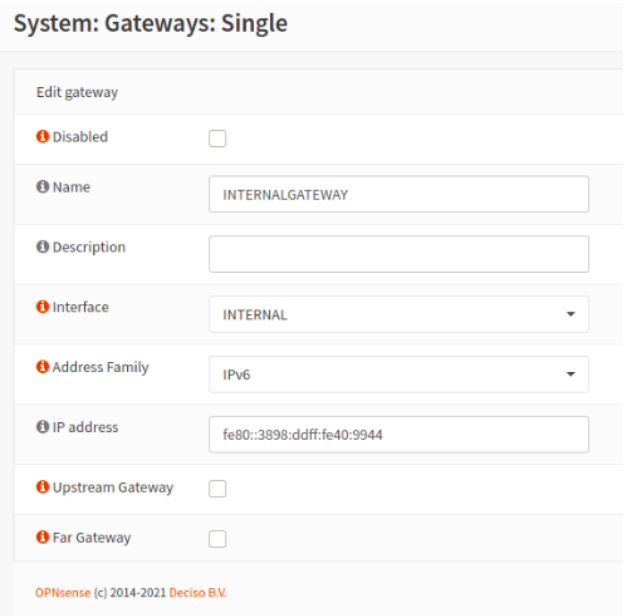
In order to maintain the network infrastructure, we have to add the routes of the routers.
The first thing we do is to allow IPv6 traffic in both routers, to do that we use the same *Virtual Machine 535* as before and after accessing the routers' interfaces, we go on *Firewall>Settings>Advanced* and we *Allow IPv6*.

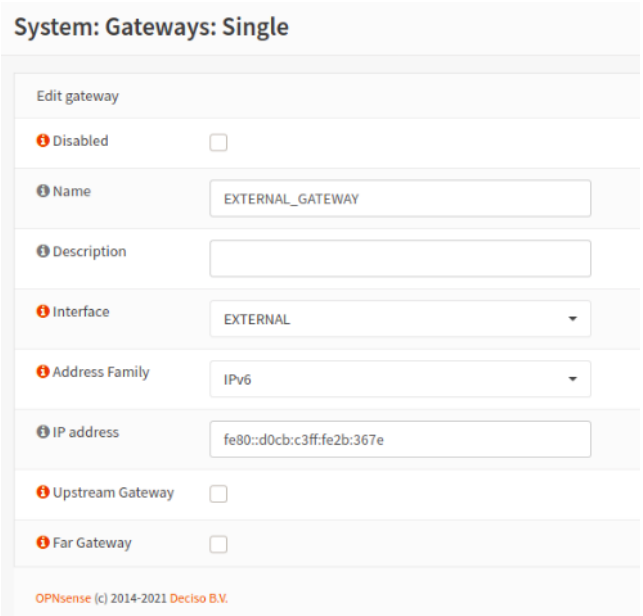


Then, we Shutdown and Start both routers in order to make the changes take place.
On *Firewall>Rules* of each router, we select each interface one by one and add a rule that allows the IPv6 traffic to flow, like for the IPv4.



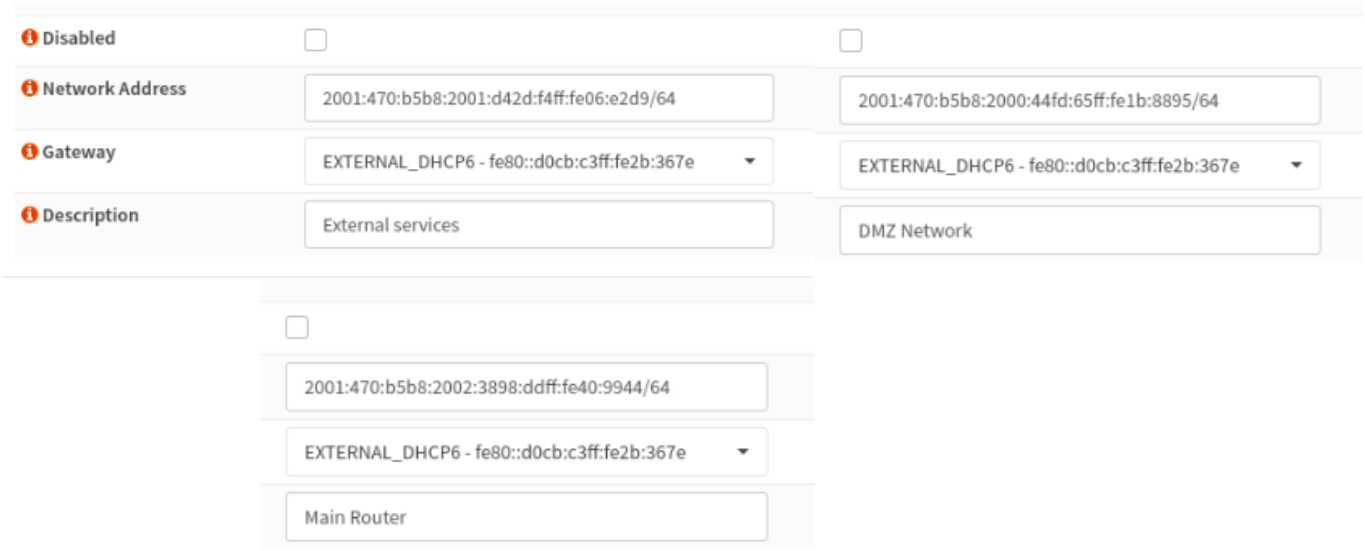
We have to set the routes and gateways to make that traffic flow after allowing it to do so.
We set an *Internal_Gateway* in the *Main Router* and an *External_Gateway* in the *Internal Router*. We set them in *System>Gateways>Single*: the IPv6 addresses are fe80::3898:ddff:fe40:9944 for the *internal gateway* and fe80::d0cb:c3ff:fe2b:367e for the *external gateway*.





We need to set the routes in *System>Routes>Configuration*; in the *Main Router* we set the routes to the *Internal Router*, the *Clients network* and the *Servers network*. And in the *Internal Router* we set them to the *Main Router*, the *DMZ* and the *External services*. The ones of the *Main Router* use the *Internal Gateway*, the ones of the *Internal Router* use the *External* one. The destination network addresses are the GUAs of the router for each subnetwork.

Internal Router:



Main Router:

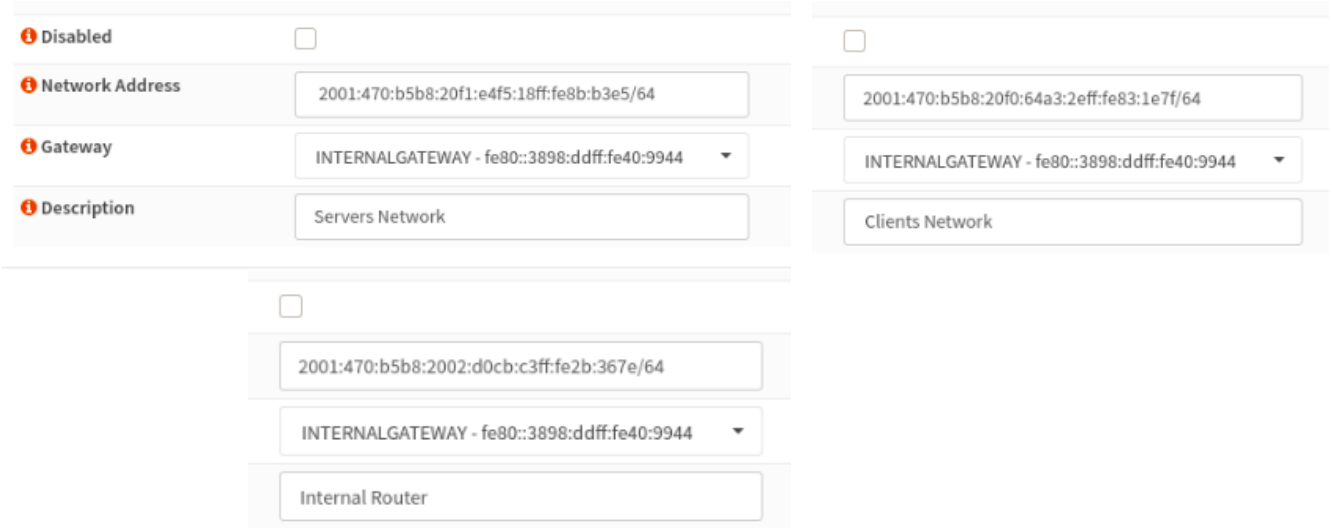


Table of the routers' GUAs:

Main Router	Interface	Global Unicast Address
	DMZ (em1)	2001:470:b5b8:2000:44fd:65ff:fe1b:8895/64
	External Services (em3)	2001:470:b5b8:2001:d42d:f4ff:fe06:e2d9/64
	Internal (em2)	2001:470:b5b8:2002:d0cb:c3ff:fe2b:367e/64

Internal Router	Interface	Global Unicast Address
	Clients Network (em2)	2001:470:b5b8:20f0:64a3:2eff:fe83:1e7f/64
	Servers Network (em1)	2001:470:b5b8:20f1:e4f5:18ff:fe8b:b3e5/64
	External (em0)	2001:470:b5b8:2002:3898:ddff:fe40:9944/64

Table of IPv4 and IPv6 addresses for each host:

Host	IPv4	IPv6
Client ext1	100.100.4.100	2001:470:b5b8:2001:e480:50ff:fe76:1546
Web Server	100.100.6.2	2001:470:b5b8:2000:40fa:57ff:fe4a:2073
Proxy Server	100.100.6.3	2001:470:b5b8:2000:8410:9bff:fe35:525c
Log Server	100.100.1.3	2001:470:b5b8:20f1:14cd:d6ff:fe00:4e4c
Kali (.100 PC)	100.100.2.100	2001:470:b5b8:20f0:e480:50ff:fe76:1546
Arpwatch	100.100.2.254	2001:470:b5b8:20f0:b89d:d3ff:feec:ea07
Domain Controller	100.100.1.2	2001:470:b5b8:20f1:4c34:16ff:fe3d:beb3

2. DNS configuration

All the internal hosts have to be able to access the internal DNS in the Server network. For this purpose we configure the DNS service in the Domain Controller (dc) machine using *dnsmasq*. *dnsmasq* is free software providing Domain Name System (DNS) caching, a Dynamic Host Configuration Protocol (DHCP) server, router advertisement and network boot features. We use it just for the DNS option, and we use it because it provides DNS service for both IPv4 and IPv6.

So the first thing we do is to go on the right machine *Virtual Machine 531* and install *dnsmasq*.

```
sudo apt install dnsmasq
```

In the dc, dnsmasq is configured by *systemd*, a software suite. This software in turn is managed by *systemctl*. So in order to use dnsmasq, we need to start it and enable it.

```
sudo systemctl start dnsmasq
```

```
sudo systemctl enable dnsmasq
```

We faced a problem here, it didn't work immediately. We did some research and we found out that there was another service that was launched by *systemctl* instead of *dnsmasq*, so what we did was *disable* and *stop* this service.

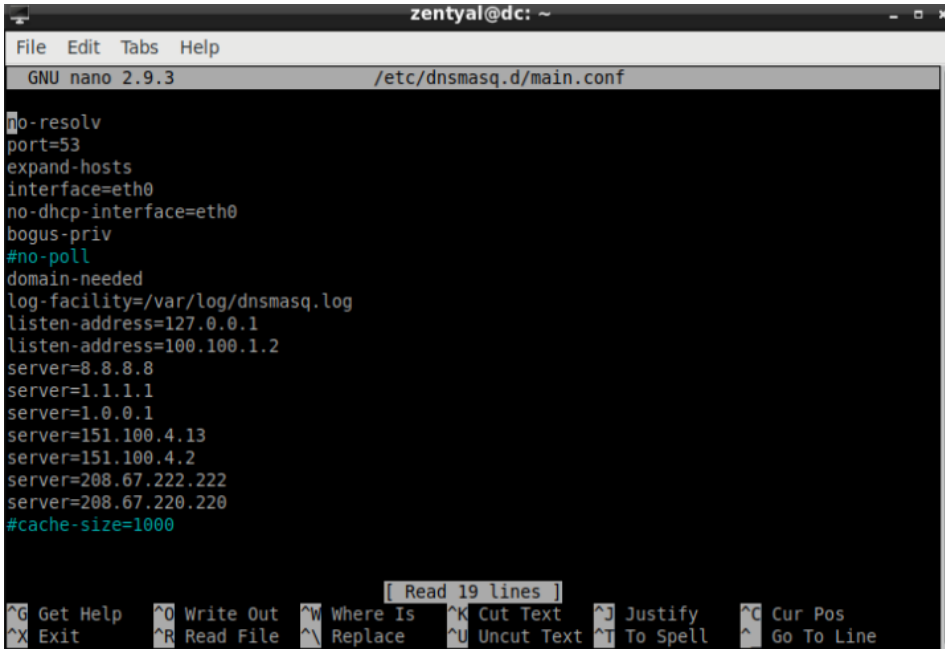
```
sudo systemctl disable systemd-resolved
```

```
sudo systemctl stop systemd-resolved
```

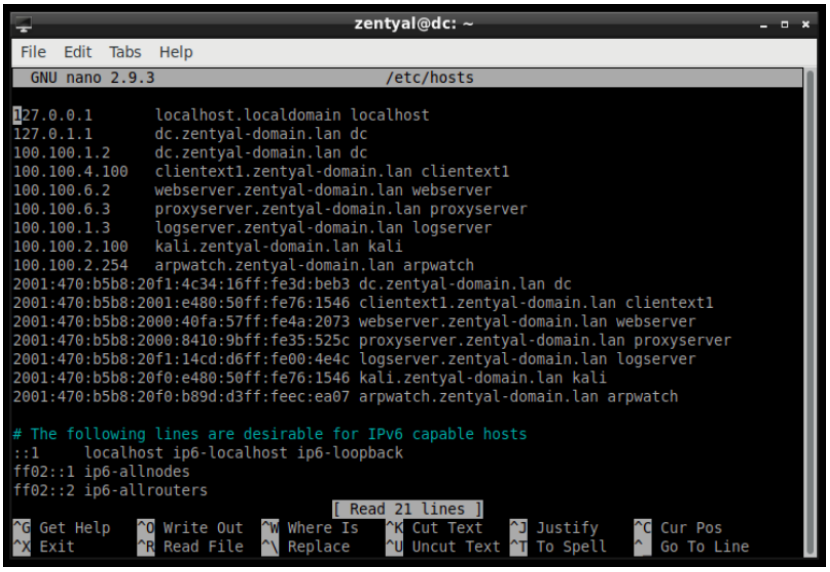
After this, dnsmasq worked. So in order to set everything up, we need to use the configuration file *dnsmasq.conf*, but instead of using this file directly, we uncommented the last line of this file that makes us use another file in the directory */etc/dnsmasq.d/*.conf* that has the *.conf* extension.
We create the file main.conf in /etc/dnsmasq.d

```
sudo nano /etc/dnsmasq.d/main.conf
```

And we write these commands:



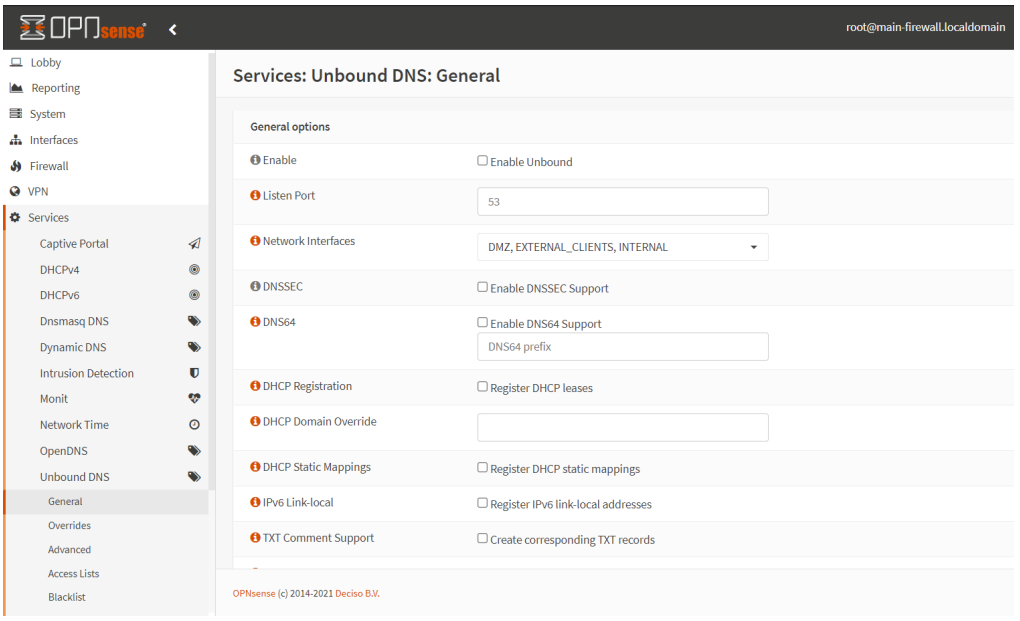
Next, we add the host we are interested to have a hostname in the */etc/hosts* file:

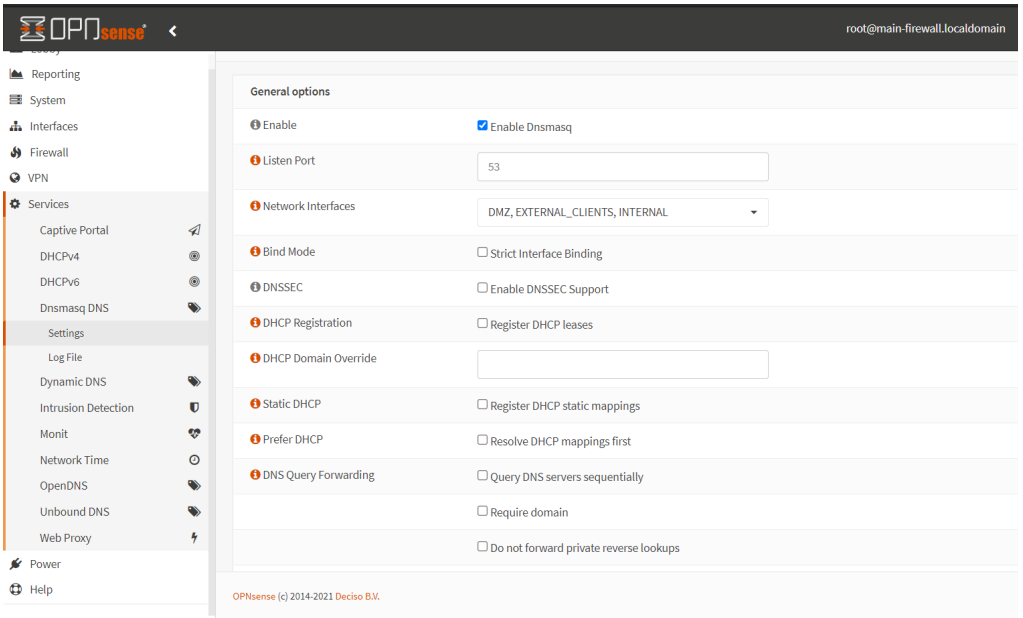


We used the domain *zentyal-domain.lan* and we set them up for both IPv4 and IPv6.
Finally, we restart the service.

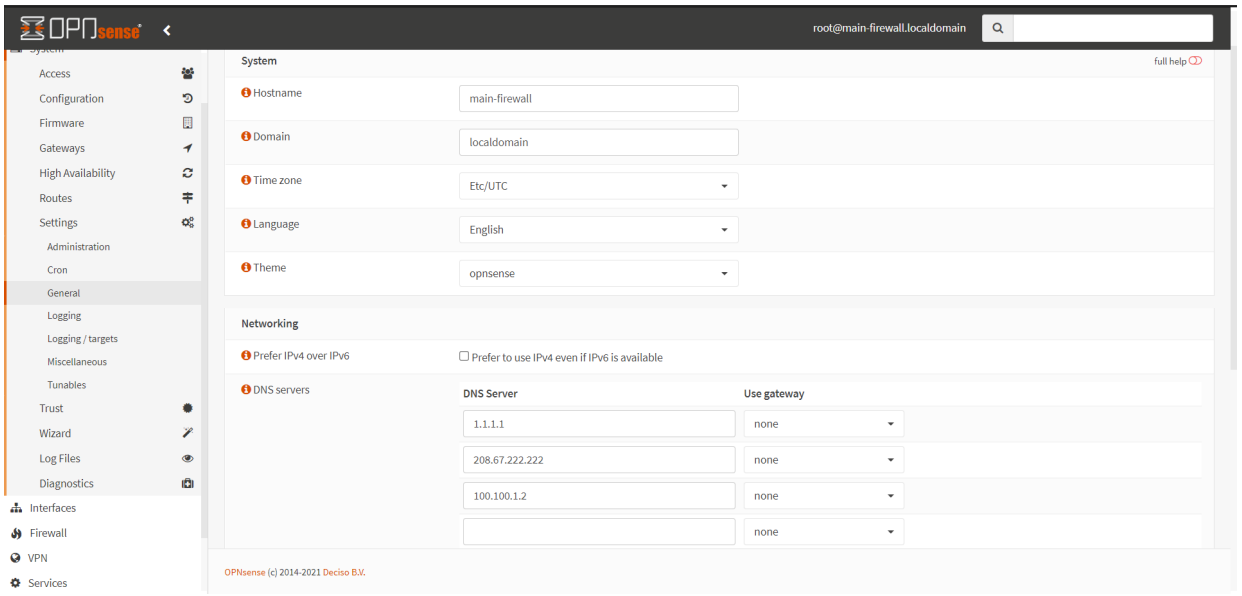
```
/etc/init.d/dnsmasq restart
```

We have to set the routers so that they accept dnsmasq and the DNS server.
On both *Main Router* and *Internal Router* we disable the *Unbound DNS* (*Services>Unbound DNS>General*)and enable the *Dnsmasq DNS* (*Services>Dnsmasq DNS>Settings*).





And we add as DNS Server the dc's IP 100.100.1.2 in *System>Settings>General>Networking>DNS servers*.



In this way, all hosts except the *Web Server*, the *Log Server* and *Arpwatch* are set to use that DNS Server. In order to set the DNS Server for these other three hosts we contact the administrators of the network.

3. Security Policy Enforcement

We have to properly implement the provided security policy, configuring the firewall rules of the *Main Firewall-Router* and the *Internal Firewall-Router*.
The rules we set are the following (we set them for both IPv4 and IPv6):

1. <u>All the hosts have to use as DNS resolver the internal DNS.</u>
   We have to set the dc machine as a DNS Resolver. We have to set the rules in our hosts, so in the EXTERNAL_CLIENTS interface of the Main Router and in the CLIENTS interface of the Internal Router. And it was set at the second point of this assignment - <u>DNS configuration</u>.

	IPv4 TCP/UDP	*	*	100.100.1.2	53 (DNS)	*	*
	IPv6 TCP/UDP	*	*	2001:470:b5b8:20f1:4c34:16ff:fe3d:beb3	53 (DNS)	*	*

2. <u>Only the webserver service provided in the DMZ has to be accessible from the Internet.</u>
   In the Main Router under DMZ, we add the rule that if the source is some host from the WAN then the destination can only be the Web Server (100.100.6.2). So we allow just the traffic from the WAN (100.100.0.2/24 and 100.101.0.0/24) to the Web Server, and if the destination is everyone else, we block it. So in the EXTERNAL_CLIENTS, CLIENTS, and SERVERS networks, we add some rules that deny the traffic from the WAN to everyone in those networks.

   In the DMZ:

▶ ← ϟ ℹ	IPv4 *	100.101.0.0/24	*	100.100.6.2	*
✖ ← ϟ ℹ	IPv4 *	100.101.0.0/24	*	*	*
▶ ← ϟ ℹ	IPv4 *	100.100.0.2/24	*	100.100.6.2	*
✖ ← ϟ ℹ	IPv4 *	100.100.0.2/24	*	*	*

In the EXTERNAL_CLIENTS, CLIENTS and SERVERS:

✖ ← ϟ ℹ	IPv4 *	100.101.0.0/24	*	*
✖ ← ϟ ℹ	IPv4 *	100.100.0.2/24	*	*

3. <u>The proxy service provided in the DMZ has to be accessible only from the hosts of the Acme network. However, the proxy needs internet access.</u>
   We set the rules in the EXTERNAL_CLIENTS, DMZ and CLIENTS network:

▶ → ϟ ℹ	IPv4 TCP/UDP	*	*	100.100.6.3	3128

In order that the proxy can access the Internet, we set these rules in the DMZ network:

▶ → ϟ ℹ	IPv4 TCP/UDP	100.100.6.3	*	*	80 (HTTP)
▶ → ϟ ℹ	IPv4 TCP/UDP	100.100.6.3	*	*	443 (HTTPS)

4. <u>All the services provided by hosts in the Internal server network have to be accessible only by Client network and DMZ hosts.</u>
   The services provided by the hosts in the Internal Server Network are the DNS on port 53 and the syslog on port 514.

▶ → ϟ ℹ	IPv4 TCP/UDP	DMZ net	*	100.100.1.2	53 (DNS)
▶ → ϟ ℹ	IPv6 TCP/UDP	DMZ net	*	2001:470:b5b8:20f1:4c34:16ff:fe3d:beb3	53 (DNS)
▶ → ϟ ℹ	IPv4 TCP/UDP	DMZ net	*	100.100.1.3	514
▶ → ϟ ℹ	IPv6 TCP/UDP	DMZ net	*	2001:470:b5b8:20f1:14cd:d6ff:fe00:4e4c	514

▶ → ϟ ℹ	IPv4 TCP/UDP	CLIENTS net	*	100.100.1.2	53 (DNS)
▶ → ϟ ℹ	IPv6 TCP/UDP	CLIENTS net	*	2001:470:b5b8:20f1:4c34:16ff:fe3d:beb3	53 (DNS)
▶ → ϟ ℹ	IPv4 TCP/UDP	CLIENTS net	*	100.100.1.3	514
▶ → ϟ ℹ	IPv6 TCP/UDP	CLIENTS net	*	2001:470:b5b8:20f1:14cd:d6ff:fe00:4e4c	514

5. <u>Anything that is not specifically allowed has to be denied.</u>
   We have just set the rules that have to be allowed, that means that everything that is not specified, is being denied.

6. <u>All the hosts (but the Client network hosts) have to use the syslog service on the Log server (syslog).</u>

The only host that we have is in the EXTERNAL_CLIENTS network.

▶ → ϟ ℹ	IPv4 TCP/UDP	EXTERNAL_CLIENTS net	*	100.100.1.3	514

7. <u>All the hosts of the network have to be managed via ssh only from hosts within the Client network.</u>

▶ → ϟ ℹ	IPv4 TCP	100.100.2.100	*	*	22 (SSH)

All the hosts in the acme network can be managed by kali (100.100.2.100) via ssh. We had a problem with the host in the EXTERNAL_CLIENTS network (100.100.4.100), it refused any port 22 connection, so what we did was to start the ssh service on that machine: *sudo service ssh start*.

8. <u>All the Client network hosts have to only access external web services (http/https).</u>

▶ → ϟ ℹ	IPv4 TCP	CLIENTS net	*	*	80 (HTTP)
▶ → ϟ ℹ	IPv6 TCP	CLIENTS net	*	*	80 (HTTP)
▶ → ϟ ℹ	IPv4 TCP	CLIENTS net	*	*	443 (HTTPS)
▶ → ϟ ℹ	IPv6 TCP	CLIENTS net	*	*	443 (HTTPS)

4. Test of the configuration

<u>IPv6 addressing</u>: we tested the set up by looking if in the Interfaces>Overview the IPv6 were there and on the single hosts we checked writing *ip -6 a* on the terminal.
<u>DNS configuration</u>: we checked if all the hosts made use of the DNS server looking inside of the */etc/resolv.conf* file.
<u>Firewall Rules</u>:

1. The same as for the DNS configuration test.
2. We pinged each host from our own terminal and it was able to receive a response only from the Web Server.
3. We checked the Internet access by accessing the *cybersecurity.uniroma1.it* site from the browser in the zentyal interface.
4. The syslog service is provided on port 514, but we cannot check if it works. The DNS using *host IP/hostname*.
5. No need.
6. Same as for point 4 (syslog part).
7. We've tested this rule by using the ssh command from kali and we were able to connect to the hosts.
8. On kali we were able to access the *cybersecurity.uniroma1.it* site from the browser, on arpwatch we used *wget* on the same site.

5. Final remarks

In the end, even if we had some problems with the IPv6 delegation and some with the dnsmasq, each host in our network has an IPv6, each one of them is able to make use of the DNS server and the traffic is limited by the use of firewall rules.