

# Homework 1 Report - Group number 27

---

Name	Student ID	email
Simone di Biasio	1823213	<a href="mailto:dibiasio.1823213@studenti.uniroma1.it">dibiasio.1823213@studenti.uniroma1.it</a>
Leonardo Persiani	1795525	<a href="mailto:persiani.1795525@studenti.uniroma1.it">persiani.1795525@studenti.uniroma1.it</a>

- [0. Brainstorming](#)
- [1. IPv6 Addressing](#)
  - [1.1 Getting the prefix](#)
  - [1.2 Subnetting](#)
    - [1.2.1 Assigning addresses to subnets](#)
    - [1.2.2 Delegating a prefix](#)
- [2. DNS Configuration](#)
  - [2.1 ACL](#)
  - [2.2 Zone File](#)
  - [2.2.3 IPv6](#)
- [3. Security policy](#)
- [4. Test of the configuration](#)
- [5. Final Remarks](#)

# 0. Brainstorming

---

After accessing the **spock** portal, the first thing we did was to change the router's default password and allow access from our host machines by adding a firewall rule, in order to easily configure them.

Our first idea was to setup the IPv6 addressing by emulating the existing IPv4 one by using static mapping, while keeping firewalls *open*. Then, configure the DNS and lastly the Security Policy.

Note: due to port forwarding on port 80 (as explained in the Security Policy section), the **OPNSense** panel is now available on port **8080**.

# 1. IPv6 Addressing

---

This section is where we struggled most: the initial idea of emulating the IPv4 mapping was not so easy as it sounds.

We thought of giving static addresses via DHCPv6 to the various servers while using **EUI64** or Random Addresses for all the other hosts which didn't require a fixed address, but we couldn't make it because:

1. while DHCPv4 uses MAC address to assign static leases, DHCPv6 uses **DUID** which is accessible only if a DHCPv6 client is running
2. the routers simply wouldn't do what we told them: we were able to see the DHCPv6 lease in the **opnsense** panel and we managed even to add a static lease, but the clients just wouldn't get the assigned address. We tried every combination of Router Advertisements+DHCPv6+dhclient but without success.

We finally went for **SLAAC** + **EUI64** addressing, which implicitly satisfied the requirement of non-random addressing for the servers.

Before proceeding with subnetting, some hosts required IPv6 to be enabled via sysctl:

```
sudo sysctl -w net.ipv6.conf.all.disable_ipv6 = 0
```

## 1.1 Getting the prefix

The first thing we did was to make sure that the Main Firewall received a **/56** prefix from the ISP router.

From the **opnsense** portal on the Main FW go to **WAN** configuration panel and set **Prefix Delegation size** to 56. Now in **WAN** interface overview we see our IPv6 GUA address and the delegated prefix:  
**2001:470:b5b8:1b00::/56**

## 1.2 Subnetting

Since our prefix is a **/56**, we decided to delegate a **/60** prefix to the internal router in order to create **/64** subnets:

- DMZ : **2001:470:b5b8:1b06::/64**
- External services : **2001:470:b5b8:1b04::/64**
- Link Net between main and internal : **2001:470:b5b8:1b07::/64**
- Delegated to Internal : **2001:470:b5b8:1b10::/60**
  - Internal Servers Network : **2001:470:b5b8:1b11::/64**
  - Clients Network : **2001:470:b5b8:1b12::/64**

### 1.2.1 Assigning addresses to subnets

The procedure is the same for every subnet, so we will take as example the **Internal** interface:

1. Set the **IPv6 Configuration Type** to **Track Interface**

2. Specify the **IPv6 Prefix ID** to (e.g. 7)
3. Apply changes

Now the hosts in the subnet will receive a Router Advertisement and they'll configure their IPs.

### 1.2.2 Delegating a prefix

To delegate a prefix we enabled DHCPv6 Server in Main Firewall on the **Internal** interface and specified the **Prefix Delegation Range** to **::10** in order to give **2001:470:b5b8:1b10::/60** to the internal router

Finally we set up the Internal Router to ask for a prefix as we did for the Main FW and assigned the corresponding addresses to interfaces.

## 2. DNS Configuration

---

Since **zentyal** does not work with IPv6, we decided to use **bind** on the Domain Controller machine for name resolution.

### 2.1 ACL

The Access Control List consists of our subnets:

```
acl "trusted" {
    localhost;

    // Internal Servers Network
    100.100.1.0/24;
    2001:470:b5b8:1b11::/64;

    // Clients Network
    100.100.2.0/24;
    2001:470:b5b8:1b12::/64;

    // External Services
    100.100.4.0/24;
    2001:470:b5b8:1b04::/64;

    // DMZ
    100.100.6.0/24;
    2001:470:b5b8:1b06::/64;

    // Link LAN
    100.100.254.0/24;
    2001:470:b5b8:1b07::/64;
};
```

This list allows us to determine which host can perform queries to our domain controller.

### 2.2 Zone File

We specified the zones for DNS lookup and Reverse DNS lookup:

```
zone "acme27.com" {
    type master;
    file "/etc/bind/db.acme27";
};

zone "100.100.in-addr.arpa" {
    type master;
    notify no;
    file "/etc/bind/db.100.100";
};
```

```
};

zone "1.b.1.8.b.5.b.0.7.4.0.1.0.0.2.ip6.arpa" {
    type master;
    notify no;
    file "/etc/bind/db.2001.470.b5b8.1b00";
};
```

and added A, AAAA and PTR records

```
[...]
log.acme27.com.      IN      A      100.100.1.3
                    IN  AAAA   2001:470:b5b8:1b11:14cd:d6ff:fe00:4e4c
[...]

[...]
3.1 IN      PTR      log.acme27.com.
[...]

[...]
c.4.e.4.0.0.e.f.f.f.6.d.d.c.4.1.1.1 IN  PTR log.acme27.com.
[...]
```

## 2.2.3 IPv6

We also enabled IPv6 to be used for query transport setting the `listen-on-v6 { any; };` option in the configuration file. As a result, queries can both be answered via IPv4 and IPv6. For example:





```
root@arpwatch-clients:~# host -6 proxy.acme27.com dc.acme27.com
Using domain server:
Name: dc.acme27.com
Address: 2001:470:b5b8:1b11:4c34:16ff:fe3d:beb3#53
Aliases:

proxy.acme27.com has address 100.100.6.3
proxy.acme27.com has IPv6 address 2001:470:b5b8:1b06:8410:9bff:fe35:525c
```

### 3. Security policy

---

We implemented the security policy in the two firewalls by keeping the default DROP policy and adding allow-type rules: the rules accurately follow the security policy and are designed to keep services running on IPv6. As an example here is the rule that allows DNS traffic in Main Firewall's External Clients interface:

	IPv4	*	*	100.100.1.2	53	*	*	All the host have to use as DNS resolver the internal DNS
	UDP				(DNS)			
	IPv6	*	*	2001:470:b5b8:1b11:4c34:16ff:fe3d:beb3	53	*	*	All the host have to use as DNS resolver the internal DNS
	UDP				(DNS)			

The implementation was pretty straightforward and we didn't find major issues in writing the rules for the requested security policy except for the specification about the DNS service. The policy stated that:

- *All the hosts have to use as DNS resolver the internal DNS*

but at the same time

- *All the services provided by hosts in the Internal Server Network have to be accessible only by Client Network and DMZ hosts*

We decided to offer DNS name resolution to the External Services' network too since it can come in handy and doesn't pose a great security concern.

Moreover, we decided to offer the HTTP service via port forwarding: all requests coming to the main firewall's IPv4 WAN address on port 80 are redirected to the webserver. Note that this means that the **OPNSense** panel is now available on port **8080**. This doesn't occur with IPv6 requests, since port forward only concerns IPv4

After the implementation, we tested the security policy by using mainly **ping**, **netcat**, **ssh** and other command line tools while monitoring in firewall logs the behavior of the two routers.

### 4. Test of the configuration

---

To test our setup we mostly used network tools like **ping**, **netcat**, **tcpdump** and the machine integrated web browser. Here's some of our tests:

- **ssh** from host in clients network towards other hosts

```

root@arpwatch-clients:~# ssh -6 zentyal@proxy.acme27.com
zentyal@proxy.acme27.com's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-136-generic x86_64)

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

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

You can access the Zentyal Web Interface at:

 * https://your_server_ip:8443

77 packages can be updated.
52 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your I

Last login: Tue Jun 22 16:43:23 2021 from 2001:470:b5b8:1b12:b89d:d3ff:feec:ea07
zentyal@proxyserver:~$

```

- HTTP connection towards web server

```

root@arpwatch-clients:~# wget http://webserver.acme27.com
--2021-06-22 15:22:06-- http://webserver.acme27.com/
Resolving webserver.acme27.com (webserver.acme27.com)... 2001:470:b5b8:1b06:40fa:57ff:fe4a:2073, 100.100.6.2
Connecting to webserver.acme27.com (webserver.acme27.com)|2001:470:b5b8:1b06:40fa:57ff:fe4a:2073|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10706 (10K) [text/html]
Saving to: 'index.html.3'

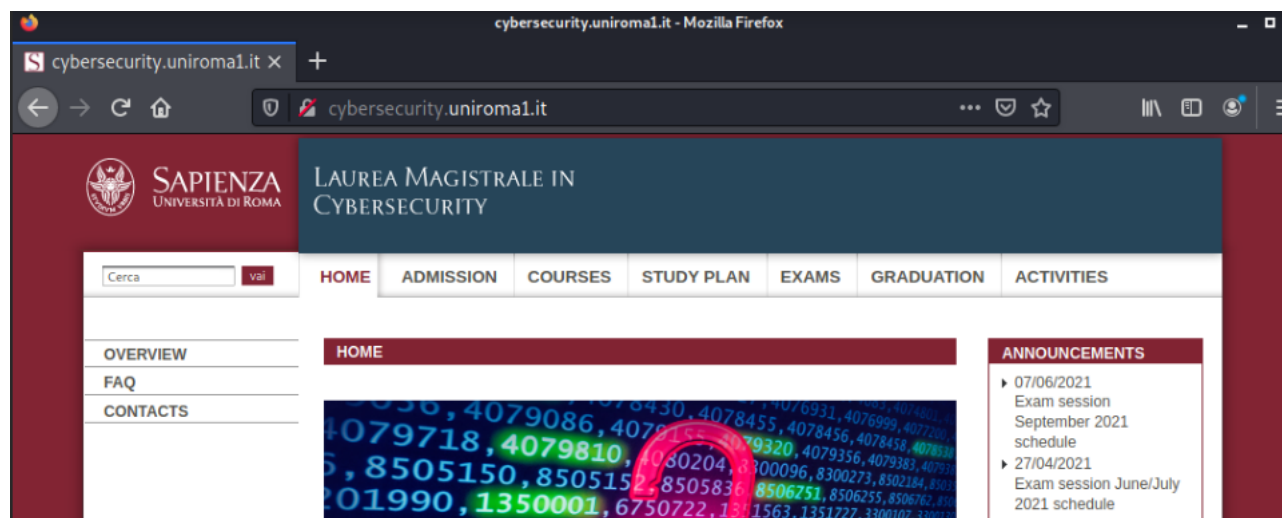
index.html.3      100%[=====] 10.46K  --.-KB/s   in 0.001s

2021-06-22 15:22:06 (11.5 MB/s) - 'index.html.3' saved [10706/10706]

root@arpwatch-clients:~#

```

- HTTP connection towards external services (from kali machine)



- Name Resolution and Reverse DNS lookup



```
root@arpwatch-clients:~# host google.com
google.com has address 142.250.180.78
google.com has IPv6 address 2a00:1450:4002:400::200e
google.com mail is handled by 20 alt1.aspmx.l.google.com.
google.com mail is handled by 10 aspmx.l.google.com.
google.com mail is handled by 50 alt4.aspmx.l.google.com.
google.com mail is handled by 30 alt2.aspmx.l.google.com.
google.com mail is handled by 40 alt3.aspmx.l.google.com.
root@arpwatch-clients:~# host webserver.acme27.com
webserver.acme27.com has address 100.100.6.2
webserver.acme27.com has IPv6 address 2001:470:b5b8:1b06:40fa:57ff:fe4a:2073
root@arpwatch-clients:~# nslookup 100.100.6.2
2.6.100.100.in-addr.arpa      name = webserver.acme27.com.
```

## 5. Final Remarks

---

We spent most of the time in trying to make IPv6 work, in particular tuning DHCPv6, Router Advertisements and Prefix Delegation in **OPNsense** panel wasn't as straightforward as we initially thought.