How to :
Configure a TP-Link router running OpenWRT to distribute 2 Internet connections to 2 SSIDs protected with WPA2-Entreprise

Henri Crombé (henri.crombe@student.uclouvain.be)
Mallory Declercq (mallory.declercq@student.uclouvain.be)

May 11, 2015

Contents

Chapter 1

Introduction

1.1 Project description

The goal of this project is to configure a TP-Link LT-WD4900 router running OpenWRT to distribute 2 WAN connections to 2 SSIDs encrypted with WPA2 Entreprise. More precisely, we would like forward traffic coming from the first SSID to the first WAN and vice-versa. Moreover, we would like our SSIDs to be protected with WPA2 Entreprise security. WPA2 Enterprise is designed for enterprise networks and requires a RADIUS authentication server. The RADIUS server will run on the router and will handle authentication requests coming from the users who wants to connect to one of the two SSIDs. Additionally, we would like the router to distribute IPv6 addresses with DHCPv6 and we would like to keep a track of which user uses which IPv6 address. The idea here is to always know which user is linked to which IPv6 addresses so that we know who does what on the network. Indeed, if we see some suspicious traffic on the network, we want to know who is the real person behind the IPv6 address.

1.2 Prerequisite

Before starting to configure your router, you have to make sure that your router is running the latest version of OpenWrt and that everything is set as default. You also have to make sure that the router you want to configure has indeed two wireless transmitter. In this document, we are working with a TP-Link WD4900 which has a 2,4 GHz and a 5 GHz transmitter. Your router should also be reachable via ssh and should be connected to the internet in order to be able to download essential packages (for the RADIUS server for example).

1.3 Situation scenario

As said previously, the goal of this project is to show how to distribute two internet connections, coming from two different ISPs, to two WiFi access point using IPv6. In the context of this project, the two ISPs advertises two different IPv6 prefixes :

- ISP1 prefix is 2001:6a8:308f:100::/64 and is mapped to VLAN 2.

- ISP2 prefix is 2001:6a8:308f:101::/64 and is mapped to VLAN 3.

These prefixes are used to reach the internet and should be used to allocate IPv6 addresses to the network we are going to build. But before deciding how to allocate the IP addresses, we need to decide how we will manage the interfaces of the router. At first sight, at least four interfaces are needed for the project. Two interfaces are needed to handle the WAN connections and two are needed to handle the wireless connections. However, we also need a LAN interface for the remaining empty ports (i.e. Those that aren't used for the WANs interfaces).

The figure above depicts the configuration we need for the router :

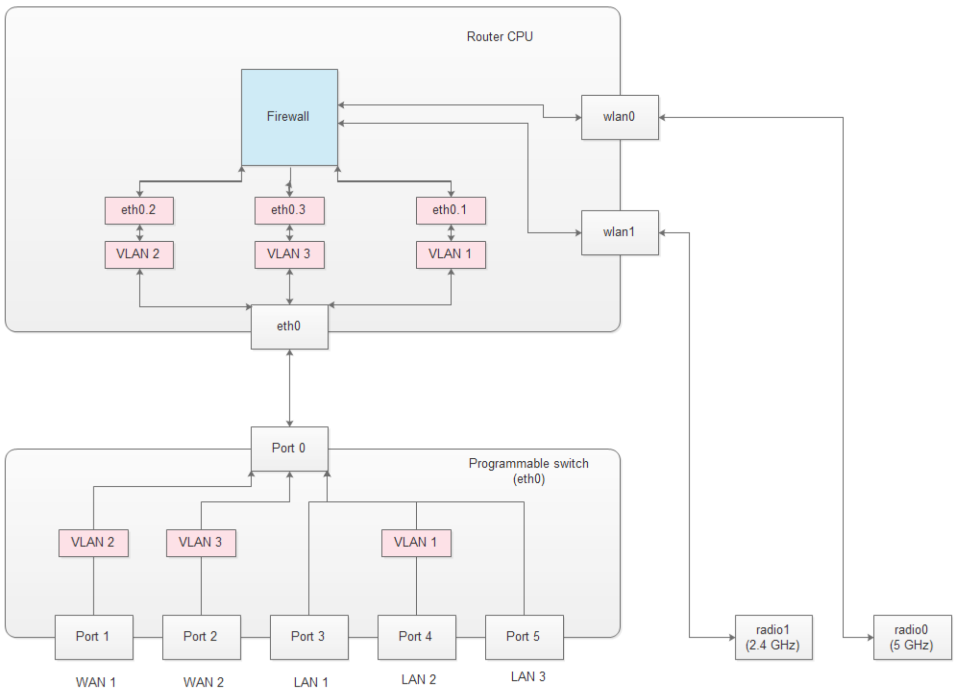


Figure 1.1: Router configuration

As you can see, default WAN port is used for WAN1 connection while the default first LAN port is used for the second WAN connection. You can also notice that the switch is handled by the interface eth0 and maps 3 different VLANs. The firewall is at the center of the configuration because it will be used to forward the traffic between interfaces.

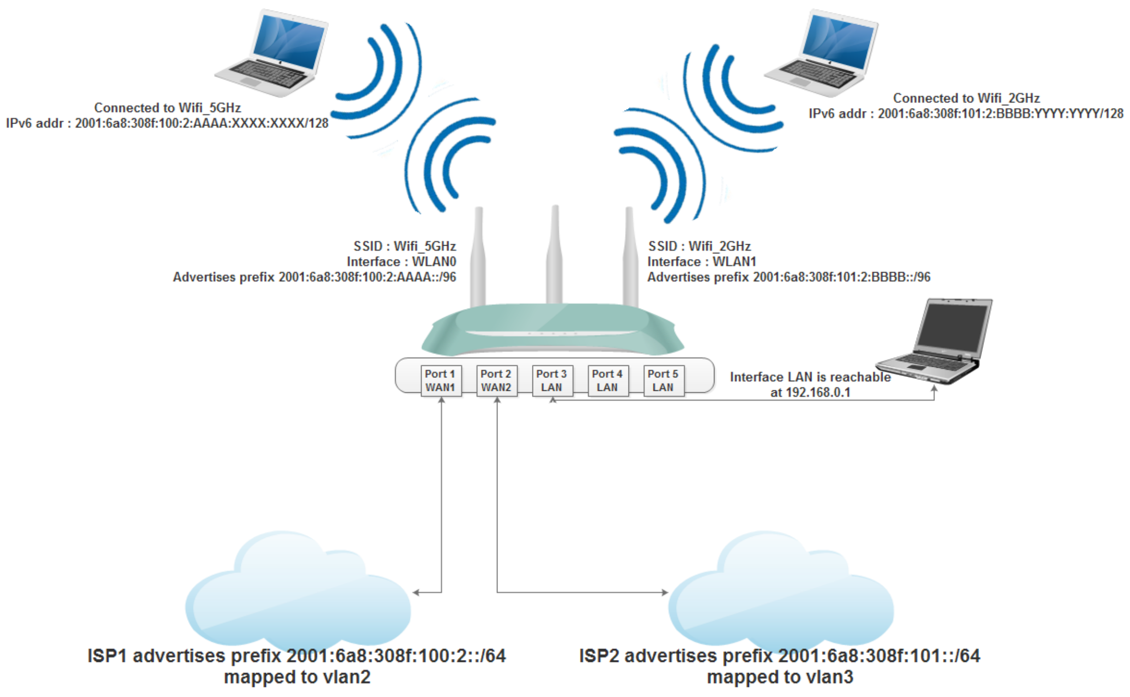
Now that we have a better understanding of the router configuration, we can consider talking about IPv6 prefixes allocation. In the context of this project, we must allocate 64 bits long prefixes for the WANs interfaces and give them static IPv6 addresses. This means that we don't rely on a DHCP to allocate IPv6 addresses to the WAN interfaces. Static addresses are also allocated to the Wlan interfaces but these interfaces advertise a longer prefix. However, we rely on a DHCP to allocate IPv6 address to the wireless clients (end-host).

For now the IPv6 allocation should look like this :

- Prefix 2001:6a8:308f:100:2::/64 is allocated to wan1.
- Prefix 2001:6a8:308f:101:2::/64 is allocated to wan2.
- Prefix 2001:6a8:308f:100:2:AAAA::/96 is allocated to wlan0.
- Prefix 2001:6a8:308f:101:2:BBBB::/96 is allocated to wlan1.

With this configuration any hosts that connect to wlan1 (Wifi_2GHz) will receive an IPv6 address that starts with the prefix 2001:6a8:308f:101:2:BBBB::/96 and connect to the internet through the second ISP (wan2). Conversely, any hosts that connect to wlan0 will receive an IPv6 address that starts with the prefix 2001:6a8:308f:100:2:AAAA::/96 and connect to the internet through the first ISP.

The diagram below show the IPv6 allocation within our network:



Connected to Wifi_5GHz
IPv6 addr : 2001:6a8:308f:100:2:AAAA:XXXX:XXXX/128
Connected to Wifi_2GHz
IPv6 addr : 2001:6a8:308f:101:2:BBBB:YYYY:YYYY/128
SSID : Wifi_5GHz
Interface : WLAN0
Advertises prefix 2001:6a8:308f:100:2:AAAA::/96
SSID : Wifi_2GHz
Interface : WLAN1
Advertises prefix 2001:6a8:308f:101:2:BBBB::/96
Port 1 WAN1
Port 2 WAN2
Port 3 LAN
Port 4 LAN
Port 5 LAN
Interface LAN is reachable at 192.168.0.1
ISP1 advertises prefix 2001:6a8:308f:100:2::/64 mapped to vlan2
ISP2 advertises prefix 2001:6a8:308f:101::/64 mapped to vlan3

Figure 1.2: IPv6 addresses allocation

Note 1: In this example, the computer connected to Wifi_5GHz has the IPv6 address :
2001:6a8:308f:100:2:AAAA:XXXX:XXXX
This means that the DHCP has allocated this address with the 96 bits long prefix of wlan0 interface and has chosen the rest of the address itself.

Note 2: The wired LAN will continues to use IPv4 addresses with DHCPv4. So, if a host plugs a link into port 3, 4 or 5, it will receive an IPv4 address and won't be able to communicate with the internet through wan1 or wan2. The idea is to always keep a reachable interface in case we mess with the network configuration. If the network configuration is messed up, you won't be able to connect to your router anymore !

The next chapter is dedicated to precisely explain how to modify OpenWrt default configuration to deploy the entire system described above.

Chapter 2

OpenWrt configuration

2.1 Required packages

At this point, we assume that your router runs the latest OpenWrt release (Barrier Breaker for now). This release of OpenWrt supports IPv6 and DHCPv6 by default. So we don't need to install any additional packages to deploy IPv6. However, we need to install some packages to deploy the RADIUS authentication server and the multi-WAN mechanism. Freeradius2 package will be used to deploy the RADIUS server and the Multiwan package will be used to handle our multi-WAN system. Next sub-sections aims to explain how to install these new packages.

2.1.1 Freeradius2

The first thing to do before installing freeradius2 is to uninstall wpad-mini because with this package the router won't be able to support WPA2. Wpad package has to be installed instead.

```
> opkg remove wpad-mini
> opkg install wpad
```

Then, to install freeradius2 package :

```
> opkg install freeradius2 freeradius2-common freeradius2
    -mod-chap freeradius2-mod-detail freeradius2-mod-eap
    freeradius2-mod-eap-md5 freeradius2-mod-eap-mschapv2
    freeradius2-mod-eap-peap freeradius2-mod-eap-tls
    freeradius2-mod-eap-ttls freeradius2-mod-exec
    freeradius2-mod-files freeradius2-mod-logintime
    freeradius2-mod-mschap freeradius2-mod-pap freeradius2
    -mod-passwd freeradius2-mod-preprocess freeradius2-mod
    -radutmp freeradius2-utils
```

Now, freeradius2 should be installed and should start when running this command :

```
> radiusd -X
```

This command should start freeradius2 in verbose mod. Freeradius2 should start normally and no errors should be displayed. The configuration of freeradius2 will be explained in the following sections.

2.1.2 Multiwan

The multiwan package is an agent script that makes Multi-WAN configuration simple, easy to use and manageable. It comes complete with load balancing, failover and an easy to manage traffic ruleset. The uci configuration file /etc/-config/multiwan is provided as part of the multiwan package.
Follow the command lines below to install and enable multiwan package:

```
> opkg update
> opkg install multiwan
> /etc/init.d/multiwan enable
> /etc/init.d/multiwan start
> /etc/init.d/multiwan single
```

Note : Multiwan will NOT work if the WAN connections are on the same subnet and share the same default gateway.
Note : Multiwan (at least on Barrier Breaker) does not accept WAN interfaces with '_' or other special characters.

2.2 Network set-up

Now that every required packages are installed, we can start to modify OpenWrt config files. Most of the configuration files are located in the folder '/etc/config/'. The list below shows which files we need to modify.

- **network :** This file is used to configure switches, interfaces and routes. This is the most important file, don't modify it unless you are sure of what you're doing. The router could become unreachable if the new config contains mistakes.
- **wireless :** This file manage wireless settings and wifi network definition (SSID, Encryption, etc..).
- **multiwan :** This file is used to handle multiwan connections.
- **dhcp :** DHCP settings, this include DHCPv6 settings.
- **firewall :** NAT, packet filter, forwarding, etc..

The next subsections are dedicated to describe how to modify these files to make the whole system run.

2.2.1 /etc/config/network

As previously said, this file is used to configure the switch and the required interfaces. The content of our network file is given below and is commented afterward.

```
config globals 'globals'
        option ula_prefix 'fd00:db80::/48'

#define loopback
config interface 'loopback'
        option ifname 'lo'
        option proto 'static'
        option ipaddr '127.0.0.1'
        option netmask '255.0.0.0'

#define interface lan (mapped to vlan 1 -> eth0.1)
config interface 'lan'
        option ifname 'eth0.1'
        option force_link '1'
        option proto 'static'
        option netmask '255.255.255.0'
        option macaddr '64:66:b3:f6:bb:00'
        option ipaddr '192.168.0.1'

#define wlan0 interface with static IPv6 address.
#wlan0 will advertise prefix 2001:6a8:308f:100:2:aaaa::/96
config interface 'wlan0'
        option proto 'static'
        option ip6addr '2001:6a8:308f:100:2:aaaa::/96'
        option macaddr '64:66:b3:f6:bb:aa'

#define wlan1 interface with static IPv6 address.
#wlan1 will advertise prefix 2001:6a8:308f:101:2:bbbb::/96
config interface 'wlan1'
        option proto 'static'
        option ip6addr '2001:6a8:308f:101:2:bbbb::/96'
        option macaddr '64:66:b3:f6:bb:bb'

#define wan1 interface with static IPv6 address
#wan1 will advertise prefix 2001:6a8:308f:100:2::/64
#the gateway for vlan 2 is 2001:6a8:308f:100::2
#mapped to vlan 2 -> eth0.2
config interface 'wan1'
        option ifname 'eth0.2'
        option proto 'static'
        option ip6addr '2001:6a8:308f:100:2::/64'
        option ip6gw '2001:6a8:308f:100::2'
        option dns '2001:4860:4860::8888'
        option macaddr '64:66:b3:f6:bb:11'

#define wan2 interface with static IPv6 address
#wan1 will advertise prefix 2001:6a8:308f:101:2::/64
#the gateway for vlan 3 is 2001:6a8:308f:101::3
```

```
#mapped to vlan 3 -> eth0.3
config interface 'wan2'
        option ifname 'eth0.3'
        option proto 'static'
        option ip6addr '2001:6a8:308f:101:2::/64'
        option ip6gw '2001:6a8:308f:101::3'
        option dns '2001:4860:4860::8888'
        option macaddr '64:66:b3:f6:bb:22'

#create the switch and enable vlan
config switch
        option name 'switch0'
        option reset '1'
        option enable_vlan '1'

#config vlan 1 on switch0
#port 3,4 and 5 are mapped to vlan 1
config switch_vlan
        option device 'switch0'
        option vlan '1'
        option ports '0t 3 4 5'

#config vlan 2 on switch0
#port 1 (default WAN port) is mapped to vlan 2
config switch_vlan
        option device 'switch0'
        option vlan '2'
        option ports '0t 1'

#config vlan 3 on switch0
#port 2 is mapped to vlan 2
config switch_vlan
        option device 'switch0'
        option vlan '3'
        option ports '0t 2'
```

As you can see, the switch configuration is quiet straightforward. You just need to create a virtual switch and tell which port is mapped to which vlan number. Note that port0, is the output port of the switch that goes to the CPU/firewall of the router.

An important thing to understand in this configuration is that when we assign a static IPv6 address to an interface, we also tell the length of the prefix to advertise. This helps OpenWrt to automatically create the correct routes between different interfaces.
Example :
-> option ip6addr '2001:6a8:308f:101:2:bbbb::/96'
This line tells OpenWrt to assign the IPv6 address 2001:6a8:308f:101:2:bbbb:0:0/128 to wlan1 interface. And it also tells the router that the interface wlan1 advertises prefix 2001:6a8:308f:101:2:bbbb::/96 to the downstream interfaces (i.e. Wifi clients).

2.2.2 /etc/config/wireless

This file is used to manage wireless settings and wifi networks (SSID, channel, Encryption, etc..). For the need of our project, the wireless file should look like this :

```
# Wireless settings

# radio0 is the 5GHz transmitter
config wifi-device 'radio0'
        #use default options ...

# radio0 device is linked to wlan0 interface.
# the wifi SSID is 'Wifi_5GHz'
# the wifi is encrypted with WPA2 Entreprise.
# client authentications are handle by RADIUS server.
# RADIUS server is hosted on localhost:1812
# the secret used to communicate with the RADIUS server is '
    testing123'
config wifi-iface
        option device 'radio0'
        option network 'wlan0'
        option mode 'ap'
        option ssid 'Wifi_5GHz'
        option encryption 'wpa2+ccmp'
        option auth_server '::1'
        option auth_port '1812'
        option auth_secret 'testing123'

# radio0 is the 2.4GHz transmitter
config wifi-device 'radio1'
        #use default options ...

# radio1 device is linked to wlan1 interface.
# the wifi SSID is 'Wifi_2.4GHz'
# the wifi is encrypted with WPA2 Entreprise.
# client authentications are handle by the same RADIUS
    server as wlan0.
config wifi-iface
        option device 'radio1'
        option network 'wlan1'
        option mode 'ap'
        option ssid 'Wifi_2GHz'
        option encryption 'wpa2+ccmp'
        option auth_server '::1'
        option auth_port '1812'
        option auth_secret 'testing123'
```

The wireless config doesn't require much explanation. As you can see, we have configured our WLAN interfaces to use WPA2 enterprise with a RADIUS server hosted on localhost. The shared secret used to communicate with the RADIUS server is set with the option 'auth_secret'. Note that different types of encryption are available, refer to the OpenWrt Wiki if you want to choose another one.

2.2.3 /etc/config/multiwan

The following file is used to configure the multiwan mechanism. By default multiwan package makes uses of load-balancing and fail over management. However, we won't rely on load-balancing or any other mechanism to route the traffic going through both WANs. So, if wan1 link fails, clients connected to Wifi_5GHz won't be able to reach the internet anymore (and vice-versa). Please note that multiwan is not in charge of routing any prefixes or traffic. It is the firewall's job to do the traffic forwarding.

```
# Multiwan configuration file

config 'multiwan' 'config'
        # REMOVE THIS LINE OR PUT TO 1 TO ENABLE MULTIWAN
        option 'enabled' '1'
        # The default route doesn't mean that the traffic is
             routed to wan2.
        # The firewall is in charge of routing.
        option 'default_route' 'wan2'
        option 'debug' '1'

config 'interface' 'wan1'
        option 'weight' '10'
        option 'health_interval' '10'
        option 'icmp_hosts' 'dns'
        option 'timeout' '1'
        option 'health_fail_retries' '2'
        option 'health_recovery_retries' '2'
        option 'failover_to' 'disable'

config 'interface' 'wan2'
        option 'weight' '10'
        option 'health_interval' '10'
        option 'icmp_hosts' 'dns'
        option 'timeout' '1'
        option 'health_fail_retries' '2'
        option 'health_recovery_retries' '5'
        option 'failover_to' 'disable'
```

As you can see, the option 'failover_to' is set to disable for both WANs. This means that there is no interface to take the relay if link fails.
If you want more information about multiwan and load-balancing, we suggest you to visit this webpage :
http://wiki.openwrt.org/doc/uci/mutliwan

2.2.4 /etc/config/dhcp

The dnsmasq and dhcpd configuration is located in /etc/config/dhcp and controls both DNS and DHCP server options on the router. To enable the DHCP to assign IPv6 addresses to wireless clients, we need to add these kind of lines to the dhcp config file :

```
config dhcp 'wlan0'
        option interface 'wlan0'
        option dhcpv6 'server'
  option ra 'server'
        list dns '2001:4860:4860::8888'
```

The first option tells the DHCP to automatically assign IPv6 addresses to clients connecting to Wifi_5GHz. The second option enables IPv6 assignments. The third one enables router advertisement for downstream interfaces. And the last option is used to advertise the main DNS IP address.
This being said, here is what your dhcp config file should look like :

```

#DHCPv4 config
config dnsmasq
        #default options...
        option leasefile '/tmp/dhcp.leases'
        option resolvfile '/tmp/resolv.conf.auto'

#DHCPv6 config
config odhcpd 'odhcpd'
        option maindhcp '0'
        option leasefile '/tmp/hosts/odhcpd'
        option leasetrigger '/usr/sbin/odhcpd-update'

config dhcp 'lan'
        option interface 'lan'
        option start '2'
        option limit '100'
        option leasetime '24h'

config dhcp 'wlan0'
        option interface 'wlan0'
        option dhcpv6 'server'
  option ra 'server'
        list dns '2001:4860:4860::8888'

config dhcp 'wlan1'
        option interface 'wlan1'
        option dhcpv6 'server'
        option ra 'server'
        list dns '2001:4860:4860::8888'

config dhcp 'wan1'
        option interface 'wan1'
        option ignore '1'

config dhcp 'wan2'
```

```
        option interface 'wan2'
        option ignore '1'
```

Note : The option 'ignore' is used to specify whether the DHCP should assign or not any IP address to this interface.
Note : The options 'dhcpv6' and 'ra' has three possible values. These values specifies whether the DHCPv6 server should be enabled (server), relayed (relay) or disabled (disabled).
Note : DHCP6 logs are stored in the file /tmp/hosts/odhcpd. Each time the DHCP assigns an IPv6 address, it updates its log file. It can be very useful if you want to know which IPv6 address has been assigned to which MAC address.

2.2.5 /etc/config/firewall

The OpenWrt Firewall provides a configuration interface that abstracts from the iptables system to provide a simplified configuration model that is fit for most regular purposes while enabling the user to supply needed iptables rules on his own when needed. The firewall maps two or more Interfaces together into Zones that are used to describe default rules for a given interface, forwarding rules between interfaces, and extra rules that are not covered by the first two. So, the firewall must be configured in order to correctly forward traffic coming from all interfaces. In our context, we would like to forward traffic coming from both wlan0 and wlan1 interfaces to wan1 and wan2 respectively.

First, we need to create a zone for each interfaces. The result should look like this :

```
# Firewall config file

config defaults
        option syn_flood        1
        option input            ACCEPT
        option output           ACCEPT
        option forward          ACCEPT

config zone
        option name             lan
        list   network          'lan'
        option input            ACCEPT
        option output           ACCEPT
        option forward          ACCEPT

config zone
        option name             wlan0
        option network          'wlan0'
        option input            ACCEPT
        option output           ACCEPT
        option forward          ACCEPT

config zone
        option name             wlan1
        option network          'wlan1'
        option input            ACCEPT
        option output           ACCEPT
        option forward          ACCEPT

config zone
        option name             wan1
        list   network          'wan1'
        option input            REJECT
        option output           ACCEPT
        option forward          REJECT
        option masq             1
        option mtu_fix          1

config zone
        option name             wan2
        option network          'wan2'
        option input            REJECT
        option output           ACCEPT
        option forward          REJECT
        option masq             1
        option mtu_fix          1
```

Then, the forwarding rules has to added. These lines will configure the firewall to forward the traffic coming from the WLANs to the corresponding WANs. This will automatically assigns the correct routes for those interfaces.

```
config forwarding
        option src    wlan1
        option dest   wan2

config forwarding
        option  src   wlan0
        option  dest  wan1
```

2.3 RADIUS authentication server

2.3.1 Why a RADIUS server ?

As you already know, we want a strong encryption and an authentication system for the two Wifi networks. This can be done by coupling the WPA2 Enterprise encryption with a RADIUS server. The system is quiet simple, every user that wants to connect either of the two Wifi has to enter a login and password. These credentials are used to identify a unique person and every connection attempts are logged into a file. If the authentication is successful, the server generates keying-materials for the encrypted wireless connection between the client and the AP. And that ends the RADIUS server's involvement.

Another reason why we choose to use an authentication server is that it records every connection attempts. And by doing so, it records the user credentials with its MAC address. This gives the opportunity, with the help of the DHCP leases, to keep a track of which IPv6 address has been assigned to which user according to the MAC address.

2.3.2 RADIUS server set-up

If you carefully followed the instructions, you should already have install freeradius2 package and it should start normally when you enter this command :

```
> radiusd -X
```

Now that we are sure that freeradius2 starts without problems, we can modify the default set-up to suit our needs. Go to the folder '/etc/freeradius2/'. You should see many files but here's which ones that are useful to us :

- **radiusd.conf :** is the main configuration file of freeradius2.
- **clients.conf :** is used to configure how and which clients can communicate with the RADIUS server. In this case, the clients are the Access Point interfaces (i.e. wlan0 and wlan1).
- **users :** is the file containing authentications rules for end-users.

/etc/freeradius2/users

As said above, this file containing authentications rules for end-users. We need to add at least one user in this file in order to connect to one of the two Wifi. This means that only one file is used for the users of both Wifi. A user is identified by a login and a password. We can apply different rules for each users (ban,allow,groups,etc). For example, if you want the user bob to be part of the list of authorized clients, just add this line to the users file :

```
bob    Cleartext-Password := "bobspasswd"
```

/etc/freeradius2/clients.conf

This file is used to manage how the access points reach the RADIUS server. The server will run on localhost and will listen to port 1812. This is the default configuration and we will use it that way. However, the line defining the shared secret between the two APs and the RADIUS server have to be changed.

```
secret = 'testing123'
```

Remember that we chose this shared secret when configuring both Wifi interfaces :

```
# In file : /etc/config/wireless
option auth_secret 'testing123'
```

Moreover, since wlan0 and wlan1 interfaces only uses IPv6 addresses, you need to set-up the RADIUS server to use IPv6 addressing:

```
#ipaddr = 127.0.0.1      # comment this line
ipv6addr = ::1           # uncomment this line
```

/etc/freeradius2/radiusd.conf

As in the clients file, you need to set-up the RADIUS server to use IPv6 addressing. You only need to comment and uncomment these lines :

```
#ipaddr = 127.0.0.1      # comment this line
ipv6addr = ::1           # uncomment this line
```

The log section is interesting since it offer the possibility to manage the output content of the RADIUS log file. From now, the server should be correctly configured to answer authentication requests coming from both Wifi.

2.4 Put it all together

The last step to make the whole system work autonomously is to update the file '/etc/rc.local' in order to start the freeradius2 and multiwan modules at the system boot.
To do so, just add the following lines into the file :

```
radiusd start
multiwan start
exit 0
```

From this moment, every time the router boots, all the required modules should start with it and everything should work as planned without having to do anything.

Bibliography

[1] OpenWrt documentation, http://wiki.openwrt.org/doc/uci

[2] OpenWrt doc : network config, http://wiki.openwrt.org/doc/uci/network

[3] OpenWrt doc : dhcp config, http://wiki.openwrt.org/doc/uci/dhcp

[4] OpenWrt doc : firewall config, http://wiki.openwrt.org/doc/uci/firewall

[5] OpenWrt doc : wireless config, http://wiki.openwrt.org/doc/uci/wireless

[6] OpenWrt doc : multiwan config, http://wiki.openwrt.org/doc/uci/multiwan

[7] OpenWrt doc : IPv6 network config,
http://wiki.openwrt.org/doc/uci/network6

[8] OpenWrt dnsmasq : dhcp config, http://wiki.openwrt.org/doc/howto/dhcp.dnsmasq

[9] Freeradius config,
https://jackofallit.wordpress.com/2012/02/15/turn-a-60-120-router-into-an-enterprise-class-wireless-router-with-openwrt/

[10] Freeradius config,
http://www.blog.10deam.com/2015/01/08/install-freeradius2-on-a-openwrt-router-for-eap-authentication/

[11] Freeradius config,
http://www.tldp.org/HOWTO/8021X-HOWTO/freeradius.html