Using IPv6 – Fosscomm 2011

George Kargiotakis

Infrastructure

End Hosts: The lab is conducted in Virtual machines (VMs) running *Debian Linux 5 (lenny)*

Installed Software:

- IPv4/IPv6 dual stack
- ISC-Bind 9.x
- Apache 2.2
- OpenSSH 5.1

Network Equipment: Juniper MX960 router.

Topology: All the VMs are connected in the same subnet.

Addressing: The VMs are connected in the following network:

- IPv6 Prefix : 2001:648:2ffc:105::/64
- IPv6 gateway : 2001:648:2ffc:105::1/64

Access

You may access the root account of the VMs through *secure shell* with the following password: ***ipv6ws123***

Please type the command:

ssh root@s-**XX**.sandbox.ypepth.grnet.gr,

where «**XX**» varies from «11» to «40» according to the number that is allocated to your team from the workshop organizers.

Note: In case that you have to install a *ssh client*, *PuTTY* is suggested*:*

(http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html/)

Part A: Transition Mechanisms

You are going to try a simple tunneling transition mechanism operated by GRNET and provided by sixxs.net. Requires prior registration!

```
# aptitude install aiccu
```

Ignore all warnings and errors

Edit /etc/aiccu.conf with an editor(nano or vim or whichever you prefer)

Change the lines

```
username AAAAA
password BBBBB
protocol tic
server tic.sixxs.net
daemonize true
automatic true
```

```
# /etc/init.d/aiccu start
```

```
# ping6 www.grnet.gr
```

For the rest of the exercises we are going to use native IPv6 so lets tear down the tunnel

```
# aptitude purge aiccu
```

Part B: IPv6 address assignment

Good practice: Setup of servers in a subnet using *IPv6 stateless autoconfiguration* should be avoided. It is recommended to disable this option.

Tip: sysctl -w net.ipv6.conf.all.autoconf=0

Good practice: Since it is desirable that IPv6 addresses of the VMs are recognizable (despite the addition of 96 bits compared to IPv4 in dual-stack machines), usually the last byte of the IPv4 address is used as the last byte of the IPv6 address. This practice facilitates the network administrator but may decrease the security in the network subnet (why?).

Example:

s-10.sandbox.ypepth.grnet.gr

- IPv4: 62.217.124.**10**
- IPv6 (suggested): 2001:648:2ffc:105::**10**

First Ping

```
# ping6 ::1
PING ::1(::1) 56 data bytes
64 bytes from ::1: icmp_seq=1 ttl=64 time=0.045 ms
```

IPv6 Addressing: Option A

Addition

```
# ifconfig eth0 add 2001:648:2ffc:105::XX/64
```

If the prefix is not declared in the end (/64 here) it is considered as an /128 prefix (or 255.255.255.255 accordingly in IPv4). Remember that **XX** is the number of your team.

```
# ping6 2001:648:2ffc:105::1
PING 2001:648:2ffc:105::1(2001:648:2ffc:105::1) 56 data bytes
64 bytes from 2001:648:2ffc:105::1: icmp_seq=1 ttl=64 time=5.18 ms
```

```
# route -6 add default gw 2001:648:2ffc:105::1
```

```
# ping6 www.grnet.gr
PING www.grnet.gr(www.grnet.gr) 56 data bytes
64 bytes from www.grnet.gr: icmp_seq=1 ttl=61 time=1.69 ms
```

Deletion

```
# route -6 del default gw 2001:648:2ffc:105::1
# ifconfig eth0 del 2001:648:2ffc:105::XX/64
```

IPv6 Addressing: Option B

Addition

```
# ip -6 addr add 2001:648:2ffc:105::XX/64 dev eth0
# ip -6 route add default via  2001:648:2ffc:105::1
# ping6 www.grnet.gr
PING www.grnet.gr(www.grnet.gr) 56 data bytes
64 bytes from www.grnet.gr: icmp_seq=1 ttl=61 time=1.69 ms
```

Deletion

```
# ip -6 route del default via  2001:648:2ffc:105::1
# ip -6 addr del 2001:648:2ffc:105::XX/64 dev eth0
```

IPv6 Addressing: Option C

It is considered the best option, since it assures that the system will keep the same IPv6 address after rebooting.

You have to open the file /etc/network/interfaces, add the following lines

```
iface eth0 inet6 static
        address 2001:648:2ffc:105::XX
        netmask 64
        gateway 2001:648:2ffc:105::1
```

and reboot the system or the network interface.

Note: Be careful to run this in one line or you will lose networking connectivity.

```
# ifdown eth0 ; ifup eth0
```

Useful Information

The command

```
# ip -6 neigh show
```

will provide information (NDP table) for the rest of the hosts in the subnet. It is the corresponding command for arp (ARP table) in IPv4 networks.

Part C: Firewall Setup

Good practive:

ICMPv6 messages **must not be filtered** under no circumstances, since this would create serious problems like:

1. Router advertisements (in case they are active) will not be received from the end hosts
2. Neighbour discovery protocol will not be operative and will lose communication with the gateway or the rest of the hosts in the network

Firewall Rules

```
# ip6tables -nxvL
Chain INPUT (policy ACCEPT 374 packets, 328655 bytes)
    pkts      bytes target     prot opt in     out     source        destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
    pkts      bytes target     prot opt in     out     source        destination

Chain OUTPUT (policy ACCEPT 1314 packets, 72620 bytes)
    pkts      bytes target     prot opt in     out     source        destination
```

The *INPUT chain* contains the rules that are applied to the incoming packets and the *OUTPUT chain* contains the rules that are applied to the outgoing packets from the host. The *FORWARD chain* is only valid if the machine routes IPv6 packets IPv6 (IPv6 routing) and contains similar rules.

Good practice:

In the *FORWARD chain* we set *policy DROP* for precaution purposes in the case where routing functionality is accidentally turned on without the addition of rules in the *OUTPUT chain*. The policy defined in the *INPUT chain* depends on the policy we want to apply in the end host.

Rule Addition

```
# ip6tables -A INPUT -s ::1 -j ACCEPT
```

```
# ip6tables -A INPUT -s 2001:648:2ffc:105::YY -p tcp --dport 22 -j ACCEPT
```

Note: Where <**YY**> place the number of the previous team. If the team next to you is already done you should not be able to SSH to their VM.

In order to add the rule in a specific point (before another rule), you have to find the incremental number of the rule prior to which the new entry will be added.

```
# ip6tables -nxvL --line-numbers
```

Example: In order to add a rule prior to rule 3, the following command has to be executed:

```
# ip6tables -I INPUT 3 -s 2001:648:2ffc:105:YY -j DROP
```

Note: The above IPv6 address is random. You are free to use the one of the team next to you.

Rule Deletion

```
# ip6tables -D INPUT -s ::1 -j ACCEPT
```

```
# ip6tables -D INPUT -s 2001:648:2ffc:105::YY -p tcp --dport 22 -j ACCEPT
```

Note: The above IPv6 address is random. You are free to use the one of the team next to you.

Alternatively, you can check the incremental number of the rule that you want to delete with the following command:

```
# ip6tables -nxvL --line-numbers
```

Example: in order to deleted rule 5, the following command has to be executed:

```
# ip6tables -D INPUT 5
```

Deletion of all the Rules

```
# ip6tables -F
```

Useful Information

Since the firewall management is a hard process and is easy to make mistakes and/or end up with a policy that differs significantly between IPv4 and IPv6, the following tool may be used:

Ferm (http://ferm.foo-projects.org/)

Part D: DNS (Addition of AAAA/PTR entries)

The virtual machines have a DNS server installed that allows *queries (recursive and transfer)* only from localhost. It responds to IPv4 and has configured the following zones:

```
ipv6.sandbox and 5.0.1.0.c.f.f.2.8.4.6.0.1.0.0.2.ip6.arpa
```

This zone is arbitrary(i.e. does not really exist) and has not been *delegated* for obvious reasons.

In /etc/resolv.conf there is only the line

```
nameserver 127.0.0.1
```

forcing the end hosts to ask the localhost server.

In order to add an AAAA record, the procedure differs slightly from that in IPv4 networks. You have to open the appropriate file (/etc/bind/ipv6.sandbox), add the following lines at the end

Note: Again **XX** is the number of your team

```
s-XX            IN A 62.217.124.XX
s-XX            IN AAAA 2001:648:2ffc:105::XX
```

and execute the commands

```
# rndc reload
# ping s-XX.ipv6.sandbox
# ping6 s-XX.ipv6.sandbox
```

Similarly, reverse DNS records can be added by opening the appropriate file *(/etc/bind/ipv6.sandbox-reverse*), adding the following line

```
Y.X.0.0.0.0.0.0.0.0.0.0.0.0.0.0 IN PTR s-XY.ipv6.sandbox.
```

Note 1: Reverse IPv6 DNS address have their digits reversed. So if **XY** is 25 you need to put 5.2 where **Y.X** is.

Note 2: Please check that the file /etc/bind/named.conf.local reports the correct zone:

zone "5.0.1.0.c.f.f.2.8.4.6.0.1.0.0.2.ip6.arpa" {

executing the commands

```
# rndc reload
# host 2001:648:2ffc:105::XX
```

The answer will be something like

```
#.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.5.0.1.0.c.f.f.2.8.4.6.0.1.0.0.2.ip6.arpa domain name pointer s-XX.ipv6.sandbox.
```

Part E: Enable IPv6 in Apache

Apache v2.2 is preconfigured in order to support IPv6 and, thus, there is no need for modifications. Attention has to be put on the definition of the *IP-based Virtual Hosts* since the following syntax has to be followed:

```
<VirtualHost  [2001:648:2ffc:105::XX]:80 62.217.124.XX:80>
```

In the case of *Named-Based Virtual Hosts* there is no such problem.

Similarly, the use of brackets *«[..]»* is necessary on the definition of the IPv6 addresses that the web server has to listen. For example:

```
Listen [2001:648:2ffc:105::XX]:90
```

The configuration files are in /etc/apache2/

You will need to edit ports.conf and sites-enabled/000-default

Question: What is result of the above configuration?

Note: Restart command is /etc/init.d/apache2 reload

Part F: Enable IPv6 in Bind

Going back in bind:

In order to reply over IPv6, appropriate configuration is necessary in *bind*:

```
listen-on-v6 { any; }
```

Upon restarting, the DNS server listens on all the configured IPv6 addresses.

File: /etc/bind/named.conf.options

Restart command: rndc reload

If you want to listen to one or more specified addresses, “any” has to be replaced with the IPv6 addresses separated by ";". It is important to note that the following line has to be present, since otherwise ACLs are not adhered:

```
match-mapped-addresses yes;
```

The reason for this, is that for minimizing used *sockets,* most operating systems (excluding BSD) may use the same socket in order to serve IPv4 and IPv6 requests in IPv6 sockets (not the reverse). Then, the application receives requests with *ipv4-mapped-ipv6 addresses,* for example:

```
::ffff:62.217.124.210
```

Another solution is to set the *net.ipv6.bindv6only=1* variable system-wide although it may cause problems in some applications (mostly written in java).

```
# sysctl -w net.ipv6.bondv6only=1
```

Part G: Advanced issues

To monitor all IPv6 packets coming to an interface:

```
# aptitude install tcpdump
# tcpdump -vv -ni eth0 -s0 ip6
```

To block Router Advertisements for a specific interface:

```
# sysctl -w net.ipv6.conf.eth0.accept_ra=0
```

to unblock

```
# sysctl -w net.ipv6.conf.eth0.accept_ra=1
```

To disable IPv6 on an interface (newer kernels):

```
# sysctl -w net.ipv6.conf.eth0.disable_ipv6=0
```

To enable it again:

```
# sysctl -w net.ipv6.conf.eth0.disable_ipv6=1
```

To enable IPv6 forwarding on an interface:

```
# sysctl -w net.ipv6.conf.eth0.forwarding=1
```

To use temporary addresses (IPv6 Privacy extensions):

```
# sysctl -w net.ipv6.conf.eth0.use_tempaddr=2
```

Part H: Autoconfiguration

To advertise your own IPv6 prefix on your Lan you need to use radvd.

```
# ip addr add 2001:db8:aaaa:XX::1/64 dev eth0
# sysctl -w net.ipv6.conf.eth0.forwarding=1
# aptitude install radvd
```

create /etc/radvd.conf with your editor:

```
interface eth0 {
        AdvSendAdvert on;
        MinRtrAdvInterval 3;
        MaxRtrAdvInterval 10;
        AdvOtherConfigFlag on;
        prefix 2001:db8:aaaa:XX::/64 {
                AdvOnLink on;
                AdvAutonomous on;
                AdvValidLifetime 86400;
                AdvPreferredLifetime 3600;
                AdvRouterAddr on;
        };
        RDNSS 2001:db8:aaaa:XX::1 {
                AdvRDNSSPreference 8;
        };
};
```

```
# radvd /etc/radvd.conf
```

then use tcpdump to verify the advertised packets.

Tell your team next to you to enable autoconfiguration on their VM:

```
# sysctl -w net.ipv6.conf.eth0.accept_ra=1
# ip addr
```

Workshop notes based on

6DEPLOY IPv6 Training Workshop (Server Labs)

6deploy

SEVENTH FRAMEWORK PROGRAMME

VMs kindly provided by

grnet

Networking Research and Education