Check Point
SOFTWARE TECHNOLOGIES LTD.

CLOUD • MOBILE • THREAT PREVENTION

WELCOME TO THE FUTURE OF CYBER SECURITY

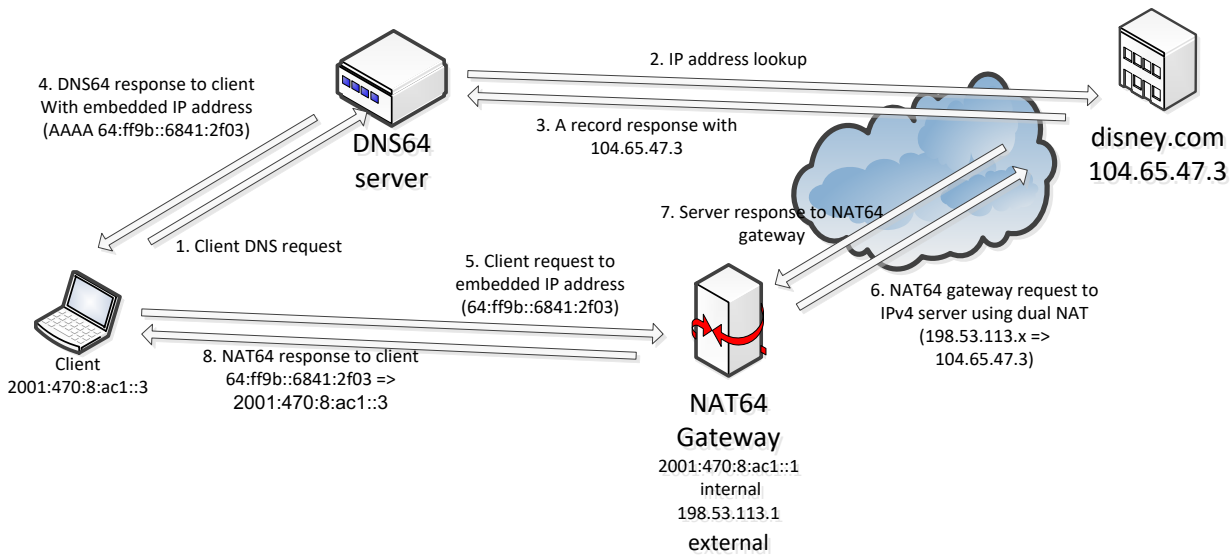
Configuring NAT64 for Internet Access in R80.20

Mark Halsall
Security Engineer
14 March 2019

Background

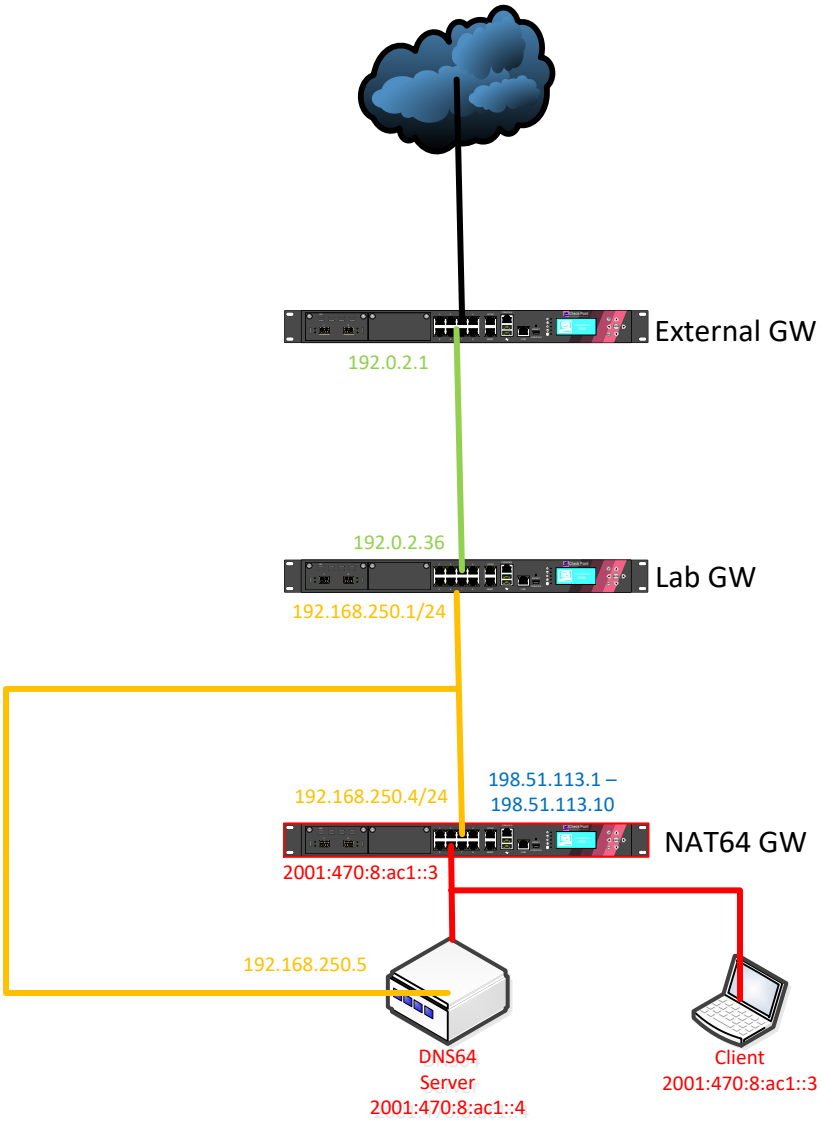
NAT64 is an IPv6 transition mechanism that facilitates communication between IPv6 and IPv4 hosts by using a form of NAT. The NAT64 gateway is a translator between IPv4 and IPv6 protocols, for which function it needs at least one IPv4 address and an IPv6 network segment comprising a 32-bit address space.

In operation, an IPv6 client makes a DNS request for the resources that the user wants. If only IPv4-based A records and not IPv6 AAAA records are returned, a DNS64 server (also required) embeds the IPv4 address the client wishes to communicate with using the host part of the IPv6 network segment, resulting in an IPv4-embedded IPv6 address (hence the 32-bit address space in the IPv6 network segment), and sends the response to the client. The client then sends packets to the resulting address. The NAT64 gateway creates a mapping between the IPv6 and the IPv4 addresses, which may be manually configured or determined automatically.



Implementation

Topology



Green gateway is an internet gateway. Not strictly necessary but used here.
Orange is the GW between ‘production’ (green) and lab networks.
Red is a dedicated IPv6 NAT64 gateway.
Blue is the address range that IPv6 client addresses are NATted to.

Procedure

These steps are required to define NAT64 rules:

1. Define a source IPv6 Network object.

This object represents the source IPv6 addresses, which you translate to source IPv4 addresses. Our example LabIPv6Net has the IPv6 subnet 2001:470:8:ac1::/96

2. Define a translated destination IPv6 Network object with an IPv4-embedded IPv6 address, or a translated destination IPv6 Host object with a static IPv6 address.

This object represents the translated destination IPv6 address, to which the IPv6 sources connect. You can also define an address range, which is what we did. The range is 64:ff9b::1 – 64:ff9b::dfff:ffff , which covers 0.0.0.1 through 223.255.255.255 (the entire non-multicast IPv4 range).

3. Define a translated source IPv4 Address Range object.

This object represents the translated source IPv4 addresses, to which you translate the original source IPv6 addresses. We used 198.51.113.1 – 198.51.113.10, which is a subset of an IPv4 address range reserved for examples.

4. Create a Manual NAT64 rule.
5. Install the Access Policy.

To define a source IPv6 Network object that represents the source IPv6 address, which you translate to source IPv4 addresses:

1. Click **Objects** menu **> New Network**.
2. In the **Object Name** field, enter the applicable name.
3. In the **Comment** field, enter the applicable text.
4. Click the **General** page of this object.
5. In the **IPv4** section:

Do not enter anything.

6. In the **IPv6** section:

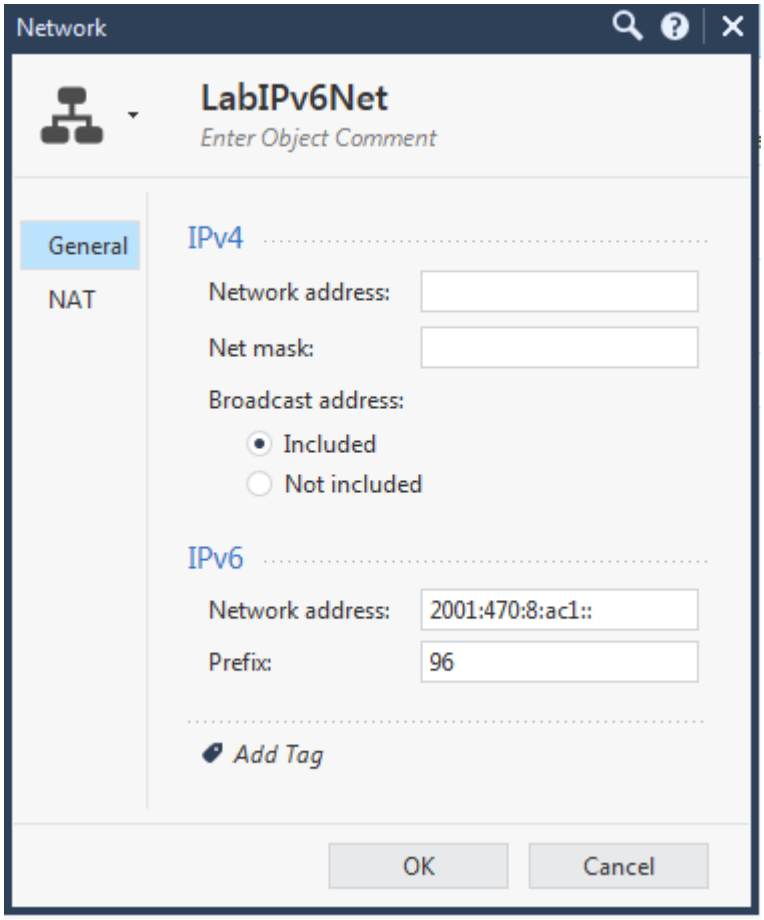
   a. In the **Network address** field, enter the IPv6 address of your IPv6 network, which you translate to source IPv4 addresses. (2001:470:8:ac1::)

   b. In the **Prefix** field, enter the prefix of your IPv6 network. (96)

7. On the **NAT** page of this object:

Do not configure anything.

8. Click **OK**.



To define a translated destination IPv6 Network object with IPv4-embedded IPv6 address that represents the IPv6 addresses, to which the IPv6 sources connect:

1. Click **Objects** menu **> New Network**.
2. In the **Object Name** field, enter the applicable name.
3. In the **Comment** field, enter the applicable text.
4. Click the **General** page of this object.
5. In the **IPv4** section:

Do not enter anything.

6. In the **IPv6** section:

   a. In the **Network address** field, enter the destination *IPv4-embedded* IPv6 address (also called *IPv4-mapped* IPv6 address), to which the IPv6 sources connect.

   RFC6052 defines the address as having the network address 64:ff9b:: and then the 32 bits of the IPv4 address in hexadecimal - 64:ff9b::WX:YZ, where WX:YZ are the four octets of the destination IPv4 address in hexadecimal format.

   For example, for IPv4 network 192.168.3.0, the IPv4-embedded IPv6 address is 64:ff9b::C0A8:0300. For more information, see RFC 6052.

   These IPv4-embedded IPv6 addresses are published by an external DNS64 server.

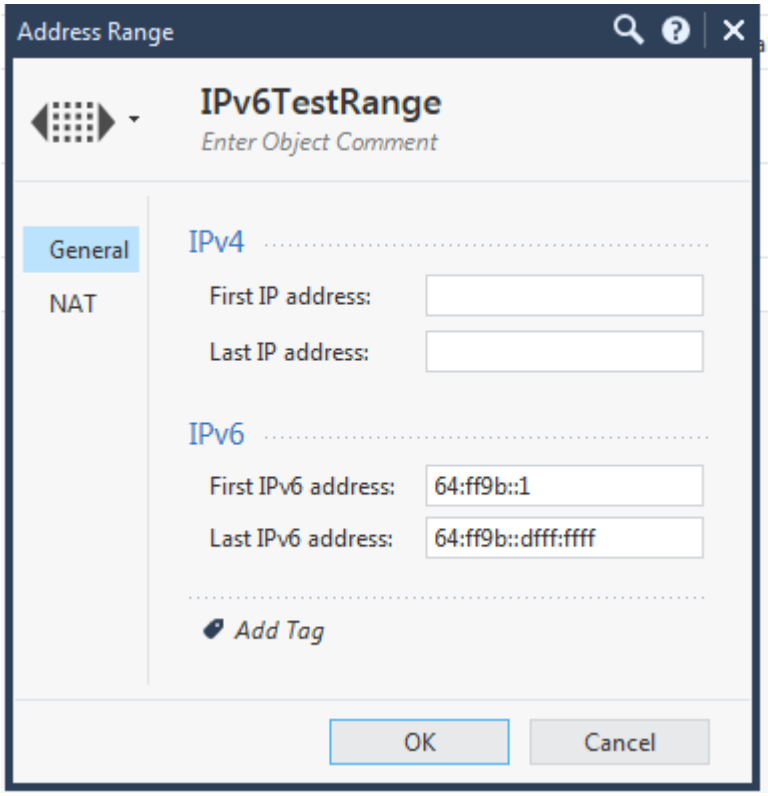
   b. In the **Prefix** field, enter the applicable IPv6 prefix.

Note - You can define IPv4-embedded IPv6 addresses only for these object types: Address Range, Network, and Host. Our example uses an Address Range.

7. On the **NAT** page of this object:

Do not configure anything.

8. Click **OK**.



To define a translated destination IPv6 Host object with static IPv6 address that represents the IPv6 address, to which the IPv6 sources connect:

1. Click **Objects** menu **> New Host**.
2. In the **Object Name** field, enter the applicable name.
3. In the **Comment** field, enter the applicable text.
4. Click the **General** page of this object.
5. In the **IPv4** section:

Do not enter anything.

6. In the **IPv6** section:

In the **Network address** field, enter the destination static IPv6 address, to which the IPv6 sources connect.

7. On the **NAT** page of this object:

Do not configure anything.

8. Configure the applicable settings on other pages of this object.
9. Click **OK**.

To define a translated source IPv4 Address Range object that represents the IPv4 addresses, to which you translate the source IPv6 addresses, follow the steps below. Our example uses the example IP range 198.51.113.1 – 198.51.113.10:

1. Click **Objects** menu **> More object types > Network Object > Address Range > New Address Range**.
2. In the **Object Name** field, enter the applicable name.
3. In the **Comment** field, enter the applicable text.
4. Click the **General** page of this object.
5. In the **IPv4** section:

   a. In the **First IP address** field, enter the first IPv4 address of your IPv4 addresses range, to which you translate the source IPv6 addresses.

   b. In the **Last IP address** field, enter the last IPv4 address of your IPv4 addresses range, to which you translate the source IPv6 addresses.

Notes:

- This IPv4 addresses range must not use private IPv4 addresses (see RFC 1918 and **Menu > Global properties > Non Unique IP Address Range**).
- This IPv4 addresses range must not be used on the IPv4 side of the network.
- We recommend that you define a large IPv4 addresses range for more concurrent NAT64 connections.

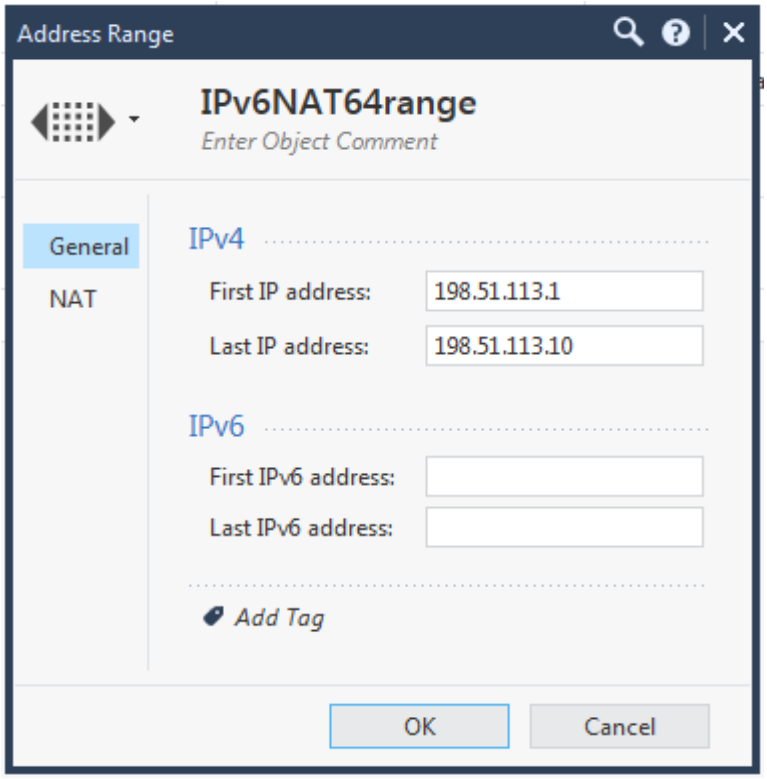
6. In the **IPv6** section:

Do not enter anything.

7. On the **NAT** page of this object:

Do not configure anything.

8. Click **OK**.



To create a Manual NAT64 rule:

1. From the left Navigation Toolbar, click **Security Policies**.

2. In the top **Access Control** section, click **NAT**.

3. Right-click on the **Manual Lower Rules** section title, and near the **New Rule**, click **Above** or **Below**.

4. Configure this Manual NAT64 rule:

Important - Some combinations of object types are not supported in the *Original Source* and *Original Destination* columns. See the summary table with the supported NAT rules at the bottom of this section.

a. In the **Original Source** column, add the IPv6 object for your original source IPv6 addresses.

In this rule column, NAT64 rules support only these types of objects:

- *Any
- Host with a static IPv6 address
- Address Range with IPv6 addresses
- Network with IPv6 address

b. In the **Original Destination** column, add a translated destination IPv6 object with an IPv4-embedded IPv6 address.

In this rule column, NAT64 rules support only these types of objects:

- Host with a static IPv6 address
- Address Range with IPv4-embedded IPv6 addresses
- Network with an IPv4-embedded IPv6 address

c. In the **Original Services** column, you must leave the default **Any**.

d. In the **Translated Source** column, add the IPv4 **Address Range** object for your translated source IPv4 addresses range.

In this rule column, NAT64 rules support only these types of objects:

- Host with a static IPv4 address, only if in the **Original Source** column you selected a Host with a static IPv6 address
- Address Range with IPv4 addresses

e. In the **Translated Source** column, right-click the IPv4 **Address Range** object > click **NAT Method** > click **Stateful NAT64**:

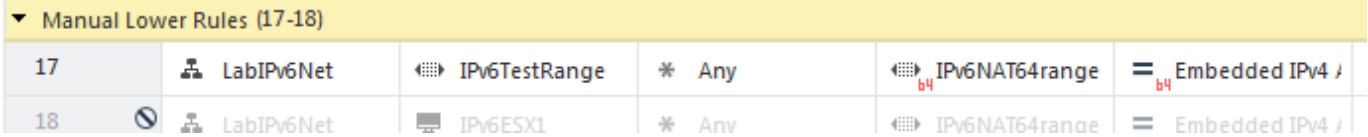
- The **Translated Packet Destination** column shows **= Embedded IPv4 Address**.
- The **64** icon shows in both the **Translated Source** and **Translated Destination** columns.

In this rule column, NAT64 rule supports only these types of objects:

- Host with a static IPv4 address, only if in the **Original Source** column you selected a Host with a static IPv6 address
- Embedded IPv4 Address

f. In the **Translated Services** column, you must leave the default **= Original**.

5. Publish the session and install the Access Policy.



LabIPv6Net has the IP 2001:470:8:ac1::/96

IPv6TestRange is 64:ff9b::1 – 64:ff9b::dfff:ffff
IPv6NAT64range is 198.51.113.1 – 198.51.113.10
Embedded IPv4 Address is autogenerated

To summarize, you must configure only these Manual NAT64 rules (rule numbers are for convenience only):

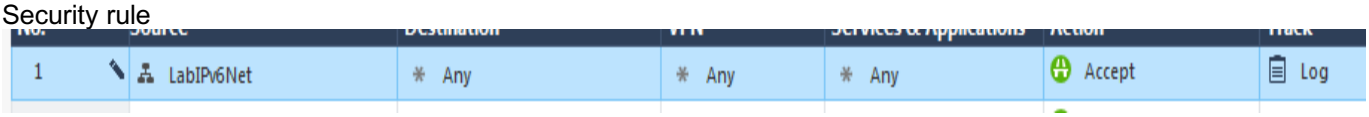
#	Original Source	Original Destination	Original Services	Translated Source	Translated Destination	Translated Services
1	*Any	IPv6 *Host* object with a static IPv6 address	*Any	IPv4 *Address Range* object	IPv4 *Host* object	= Original
2	*Any	IPv6 *Address Range* object with an IPv4-embedded IPv6 addresses	*Any	IPv4 *Address Range* object	Embedded IPv4 Address	= Original
3	*Any	IPv6 *Network* object with an IPv4-embedded IPv6 address	*Any	IPv4 *Address Range* object	Embedded IPv4 Address	= Original
4	IPv6 *Host* object with a static IPv6 address	IPv6 *Host* object with a static IPv6 address	*Any	IPv4 *Host* object	IPv4 *Host* object	= Original
5	IPv6 *Host* object with a static IPv6 address	IPv6 *Address Range* object with IPv4-embedded IPv6 addresses	*Any	IPv4 *Address Range* object	Embedded IPv4 Address	= Original
6	IPv6 *Host* object with a static IPv6 address	IPv6 *Network* object with an IPv4-embedded IPv6 address	*Any	IPv4 *Address Range* object	Embedded IPv4 Address	= Original
7	IPv6 *Address Range* object	IPv6 *Host* object with a static IPv6 address	*Any	IPv4 *Address Range* object	IPv4 *Host* object	= Original
8	IPv6 *Address*	IPv6 *Address Range*	*Any	IPv4 *Address*	Embedded IPv4	= Original

	Range object	object with IPv4-embedded IPv6 addresses		*Range* object	Address	
9	IPv6 *Address Range* object	IPv6 *Network* object with an IPv4-embedded IPv6 address	*Any	IPv4 *Address Range* object	Embedded IPv4 Address	= Original
10	IPv6 *Network* object	IPv6 *Host* object with a static IPv6 address	*Any	IPv4 *Address Range* object	IPv4 *Host* object	= Original
11	IPv6 *Network* object	IPv6 *Address Range* object with IPv4-embedded IPv6 addresses	*Any	IPv4 *Address Range* object	Embedded IPv4 Address	= Original
12	IPv6 *Network* object	IPv6 *Network* object with an IPv4-embedded IPv6 address	*Any	IPv4 *Address Range* object	Embedded IPv4 Address	= Original

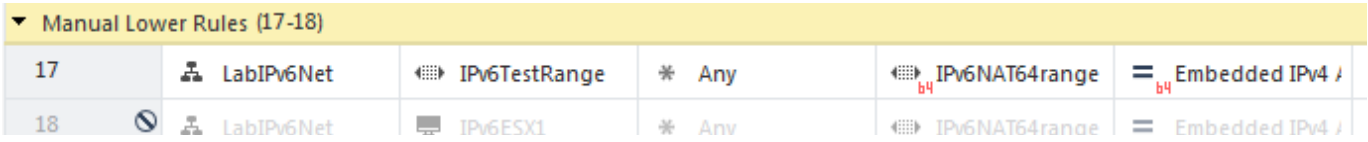
Ruleset

IPv6 gateway

Security rule



NAT rule

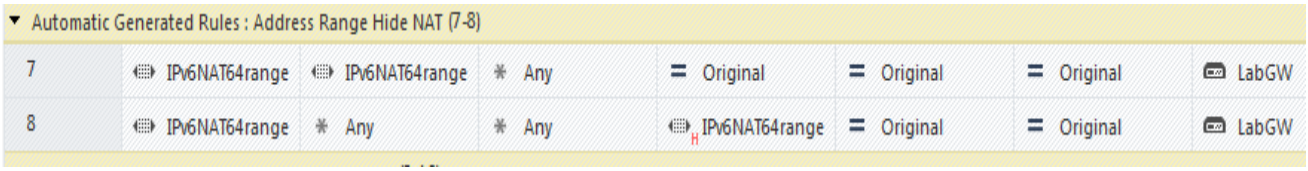


LabIPv6Net has the IP 2001:470:8:ac1::/96
IPv6TestRange is 64:ff9b::1 – 64:ff9b::dfff:ffff
IPv6NAT64range is 198.51.113.1 – 198.51.113.10
Embedded IPv4 Address is autogenerated

Lab gateway

Security rule

NAT rule



IPv6NAT64range is 198.51.113.1 – 198.51.113.10 (same object as above). We are hiding it behind the LabGW so that there aren't any adjustments needed to the 'prod' network. These IPv4 addresses have to be unused elsewhere in the configuration, and I had to set up routing and antispoofing, as well as standard hide NAT so that they had internet access.

The other big piece is a DNS server that can do DNS64. It basically takes an IPv4 address and translates it into the last 32 bits of an IPv4 address embedded in IPv6. The format is 64:ff9b::XX:XX where XX:XX is the IPv4 address translated octet-by-octet into hex. This address is passed to the client, and the NAT64 rule actually turns the source (client) IPv6 as well as the destination IPv6 addresses into IPv4 addresses. A standard, up to date bind DNS server can do DNS64 by default, and the config is very simple:

```
options {
        directory "/var/cache/bind";
forwarders {
        192.0.2.57;
};
        dnssec-validation auto;

        auth-nxdomain no;    # conform to RFC1035
        listen-on-v6 port 53 { any; };
        listen-on port 53 { any; };
        allow-query { any; };
        dns64 64:ff9b::/96 {
                clients { any; };
        };
};
```

The two most important lines are the listen-on-v6 and dns64 lines, as they enable IPv6 support and DNS64 support respectively.

