ip46nat
User's Guide

Author: Tomasz Mrugalski
version 2008-06-04

Table of contents

1 Project overview

Goal of this project is to provide solution for seamlessly forward IPv4 traffic over IPv6 networks. There are two possible migration modes: IPv4 to IPv6 NAT (see section 1.1 for details) and IPv4 over PPP over L2TP over IPv6.

1.1 Phase 1: IPv4 to IPv6 NAT

First approach to the IPv4 over IPv6 network assumes that incoming IPv4 packets will be converted and forwarded as IPv6. IPv4 addresses will be "expanded" into full IPv6 addresses, using 2 extra prefixes: M and P. Returning IPv6 packets will be converted back to IPv4.

1.2 Phase 2: PPP/L2TP

tbd

2 Project status

Linux booting on LinkSys: done

ip46nat kernel module developed: done

ip46nat module ported to LinkSys: done

prefix delegation mechanism in dibbler modified: done

dibbler port to LinkSys: work in progress (ported, crashes)

documentation: work in progress

3 Installation

To complete installation, several steps are qreuired. Following sections describe, how to achieve specific goals. In general, there are several approaches:

1. Install required software (i.e. modified Linux kernel + ip46nat module) on a PC machine and use it to perform IPv4-IPv6 NAT. This scenario may be used as an preliminary validation scenario.
2. Install Linux on LinkSys WRT54 system, install ip46nat module.

3.1 Connecting LinkSys

Before any configuration or firmware modification, make sure that you have full connectivity to your LinkSys device. LinkSys devices by default use 192.168.1.1 address, so to communicate with it, another address from 192.168.1.0/24 pool is required. For example, PC may be configured to use 192.168.1.100/24 address. To check if you have connectivity with LinkSys device, use following command:

```
ping 192.168.1.1
```

Make sure that you have LinkSys connected using the rightmost socket. See Fig.1 below.

Fig. 1: Connecting LinkSys to LAN

Note: IPv4 address set to 192.168.1.1 is a LinkSys' default setup. It reverts to this configuration after every firmware upgrade. Also it is its default factory configuration.

3.2 Firmware upgrade (using original web interface)

Before attempting to install Linux on LinkSys device, make sure that this particular model is supported. Please consult http://wiki.openwrt.org/TableOfHardware .

Linux installation on Linksys is being performed as a firmware upgrade. During the first installation, original web interface (provided by LinkSys) should be used. From PC using the same address space (see section 3.1), use web browser to connect to your LinkSys web interface. See Fig.2 below.

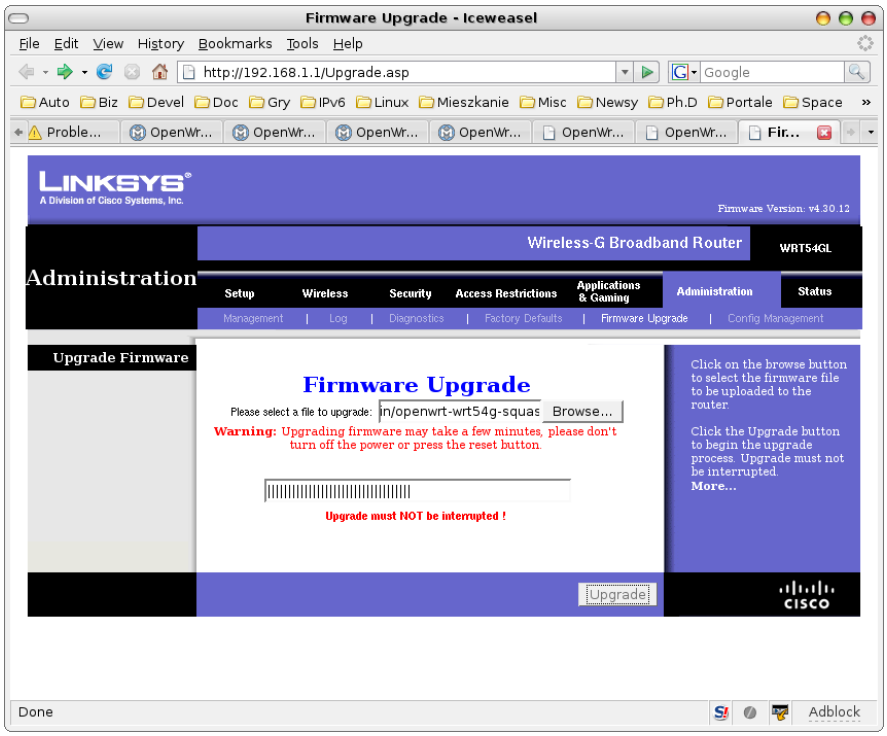

Fig. 2: Firmware upgrade using original web interface

Select appropriate firmware image (i.e. file that ends with -squashfs.bin and is corresponding to the name of used device). There are some sanity checks in the firmware upgrade procedure, but using wrong image may result in rendering the router unusable. You may want to check supported hardware list: http://wiki.openwrt.org/TableOfHardware?action=show&redirect=toh Please verify that you are using proper firmware. After firmware was uploaded, flashing takes place. It can take up to 2 minutes. After completion, router will reboot. Make sure to not reboot or power off the router before it finishes flashing.

It also may be beneficial to read following installation guide:
http://wiki.openwrt.org/InstallingWrt54gl

3.3 First connection

After performing firmware upgrade, it is possible to connect to the router using telnet command. Please run following command: telnet 192.168.1.1 from a PC console. See fig. 3 for example session. There is no root password. Please change root password by issuing passwd command. After this operation, telnet service will be disabled. SSH will be enabled instead. Note that ssh requires some time (~30 seconds) to generate keys, so it will reject any connection attempts at that time.

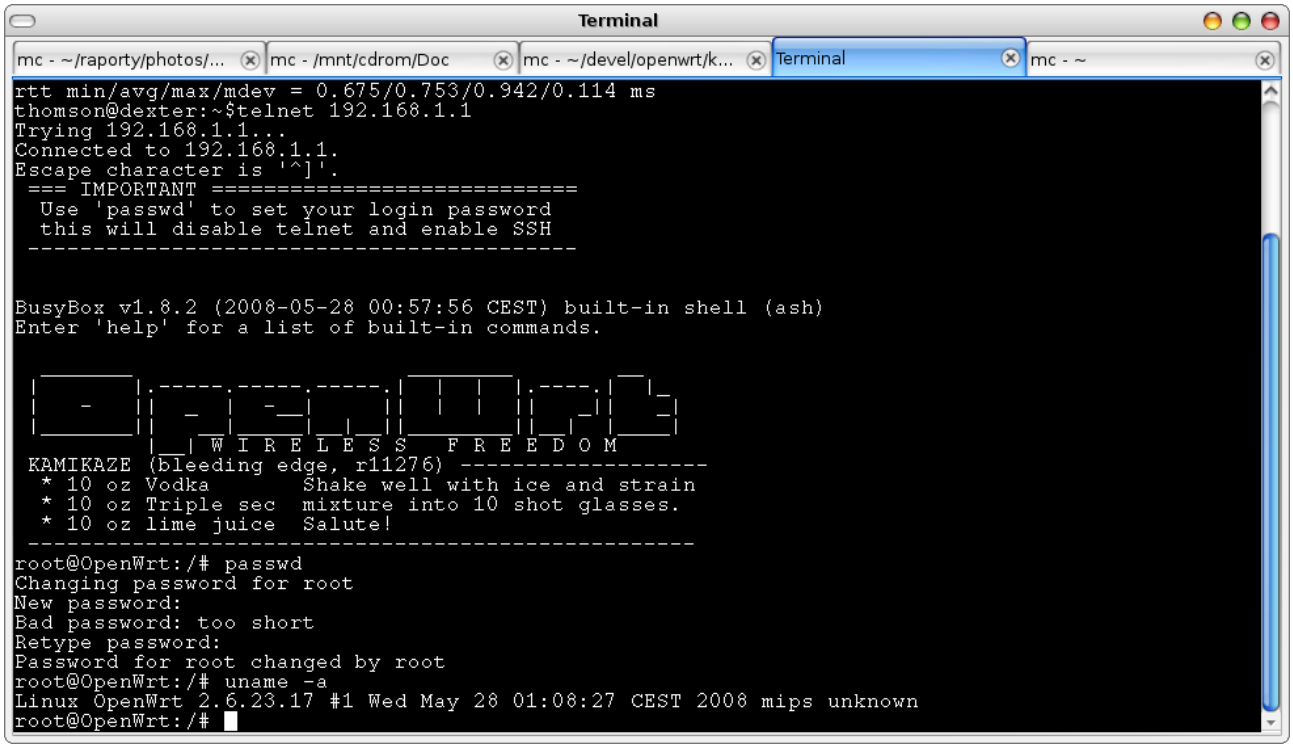

Fig. 3: Linux installed on LinkSys device

3.4 Package installation

Software installation is being done using ipk packages. To install a package please copy it to /tmp directory, e.g. using scp command (run on a PC):

```
scp mtd_6_mipsel.ipk root@192.168.1.1:/tmp
```

After package is transferred to the router, use following command:

```
ipkg install package_name.ipk
```

to install package. For example, to install mtd, use following command :

```
ipkg install mtd_6_mipsel.ipk
```

Following packages are currently recommended:

- mtd - mtd is tool used to flash router. That is the preferred way to flash router, once Linux have been installed.
- ip - powerful tool used for network configuration
- iptables - optional package, may be used to configure IPv4 firewall and/or IPv4 only NAT.
- ip6table - optional package, may be used to configure IPv6 firewall and/or IPv6 only NAT.
- uclibcxx - C++ library required to run all software written in C++, e.g. dibbler software
- dibbler-client - DHCPv6 client, used to retrieve configuration
- dibbler-server - DHCPv6 server, used to distribute addresses and configuration parameters to other nodes.

3.5 Firmware upgrade (using linux console)

To perform firmware upgrade from OpenWRT (i.e. when your LinkSys device was flashed already), mtd tool must be used. Copy openwrt-brcm47xx-squashfs.trx file to the /tmp directory. Note that this is a different image file than used in the web interface. Copy it to your LinkSys device:

```
scp openwrt-brcm47xx-squashfs.trx root@192.168.1.1:/tmp
This command will copy required firmware image to /tmp directory.
Change to that directory and begin flashing, using following
command:
```

```
cd /tmp
mtd -r write openwrt-brcm47xx-squashfs.trx linux
```

After flashing is complete, device will reboot. It takes up to 2 minutes to finish flashing and rebooting. Please note that after such firmware upgrade, all possible changes made to the router configuration will be lost. That includes all software packages installed and all configuration changes.

Note: .bin and .trx firmware image files contain the same image, but .trx is a "raw" image, while .bin has extra headers for the purpose of being recognised as a valid image by the original web interface.

3.6 Installing ip46nat kernel module

To perform IPv4-to-IPv6 NAT, a separate kernel module have been developed. For the easiness of installation, it is being distributed as a ipk package. Please install it as any other ipk packages:

```
ipkg install http://wiki.openwrt.org/InstallingWrt54gl
```

After installation is complete, ip46nat kernel module may be loaded, using following command:

```
insmod /lib/modules/2.6.23.17/ip46nat.ko v6prefixm=2000::
v6prefixp=3000:: v4addr=10.10.1.0
```

Module reports its operation using normal kernel messages. To see kernel output, dmesg command may be used. It also appears to be useful to filter dmesg output using tail command. To see last 10 messages, use following command:

```
dmesg | tail -30
```

After module is loaded, it reports opertion readiness and begins to filter incoming traffic immediately:

```
IPv4-IPv6 NAT module loaded: v6prefixp: 3000::, v6prefixm: 2000::, v4addr: 10.10.1.0
```

```
Handlers for IPv4 and IPv6 installed.
#IPv4 rcvd (rcvd so far: 1) [src=192.168.1.100, dst=192.168.1.1,looking for 10.10.1.0/24]
#IPv4 rcvd (rcvd so far: 2) [src=192.168.1.100, dst=192.168.1.1,looking for 10.10.1.0/24]
#IPv4 rcvd (rcvd so far: 3) [src=192.168.1.100, dst=192.168.1.1,looking for 10.10.1.0/24]
```

Kernel module may be unloaded at any time. To do so, use command:

```
rmmod ip46nat
```

After kernel is unloaded, statistics are being presented. To see them, use `dmesg` command.

```
Handlers for IPv4 and IPv6 removed.
---IPv4-IPv6 NAT statistics-------------------
IPv4-to-IPv6 packets: 0
IPv6-to-IPv4 packets: 0
IPv4 rcvd: 74, sent: 0
IPv6 rcvd: 0, sent: 0
IPv4 dropped (too large): 0
IPv4 dropped (no route): 0
IPv6 dropped (no route): 0
IPv4 dropped (transmission failed): 0
IPv6 dropped (transmission failed): 0
----------------------------------------------
IPv4-IPv6 NAT module unloaded.
```

Note: from the early demo perspective, it is also possible to compile and run Linux kernel with ip46nat module on a PC. For a details regarding module compilation, see section 5.

4 Network configuration

Before attempting to perform configuration, it is strongly recommended to be familiar with the following guide: http://wiki.openwrt.org/OpenWrtDocs/NetworkInterfaces

All LinkSys WRT routers use one common ethernet interface, duplicated using different vlans. In the latest OpenWRT version, that is being reported as eth0, eth0.0, eth0.1 etc.

To enable IPv4 and IPv6 forwarding, use following command:

```
echo 1 > /proc/sys/net/ipv4/conf/all/forwarding
echo 1 > /proc/sys/net/ipv4/conf/all/forwarding
```

To configure routing, ip command may be used. Assuming that we want to NAT traffic from 192.168.1.0/24 to 2000::/64, following command may be used:

```
ip route add 192.168.1.0/24 dev eth0.0
ip route add 2000::/64 dev eth0.1
```

5 NAT operation

After module is loaded (see section 3.6 for details), module will start printing information about all received IPv4 and IPv6 traffic. Packets that match configured criteria are marked using *. When match is found, packet will be recoded and transmitted.

5.1 IPv4 to IPv6 NAT

During module insertion, v4addr parameter is specified. It is being used as a source IPv4 address filter, used with /24 bitmask. For example, if v4addr is equal to 192.168.1.0, all incoming traffic from 192.168.1.0/24 network will be translated to IPv6. Source IPv6 address will be created as a concatenation of the P prefix (v6prefixp parameter specified during module insertion) and IPv4 address. Destrination IPv6 address will be created as a concatenation of the M prefix (v6prefixm parameter specified during module insertion). TTL field will be copied and decreased by 1.

For converted IPv6 packet to be transmitted successfully, IPv6 forwarding must be enabled. IPv6 routing must also be configured. See section 4 for details.

Please note that L4 layer (TCP, UDP, ICMP, etc.) checksums are not modified in any kind. That means that After IPv4 to IPv6 conversion, packets will not be accepted by destination router. They must be converted back to IPv4. That should not pose any concers, however, as routers are not supposed to intestigate L4 content at all. Thus packets must be converted back to IPv4 before they reach their destination.

Matched packet information will be reported in the following manner:

```
#IPv4 rcvd (rcvd so far: 3) [src=192.168.1.100, dst=192.168.1.1,looking for 192.168.1.0/24] *
```

5.2 IPv6 to IPv4 NAT

Once IPv4 packets are sent as IPv6, it is expected to receive responses. They are to be received as a IPv6 packets. Source IPv6 address will belong to the M prefix (v6prefixm parameter used during kernel module insertion) and will contain source IPv4 address embedded on 4 least significant octets. Destination IPv6 address will belong to the P prefix (v6prefixp parameter used during kernel module insertion) and will contain destination IPv4 address embedded on 4 least significant octets. If incoming IPv6 packet meets those criteria, it will be converted to IPv4 packet. Header checksum will be calculated, TTL will be decreased and packet will be sent.

For converted IPv4 packet to be transmitted successfully, IPv4 forwarding must be enabled. IPv4 routing must also be configured. See section 4 for details.

6 Compilation

tbd

6.1 Linux for LinkSys (OpenWRT)

tbd

6.2 ip46nat kernel module

tbd

6.3 ip46nat kernel module as an ipk package

tbd

6.4 dibbler software

tbd

6.5 dibbler software as an ipk package

tbd

7 Miscellaneous topics

tbd

8 Links

Following links are recommended reading:

- http://klub.com.pl/ip46nat/ - ip46nat project homepage.
- http://openwrt.org/ - OpenWRT project website
- http://downloads.openwrt.org/kamikaze/docs/openwrt.html - OpenWRT manual

9 Contact

Author of this project can be reached using e-mail mailto:tomasz.mrugalski@gmail.com.