IPv6Now

How to install and configure
the Hexago Gateway6 Client
on Mac OS X

Version 0v3

Author: Tony Hill, IPv6Now Pty Ltd

Table of Contents

1 Introduction

We strongly recommend that you read this complete HowTo before beginning.

If you run into problems and this HowTo doesn't help, check the FAQ as well.

2 Check requirements

You need to be running Mac OS X ("Darwin" and derivatives).

You need a working connection to the Internet in order to actually use the client. You do not need to be connected to *install* or *configure* the client however.

You need a development environment on your computer - a suitable C compiler and so on. This generally means the Mac OS X Developer Tools must be installed.

You will need root access to the computer if you wish to install the client software anywhere outside your home directory, and also in order to run the client.

3 Obtain the tuntap driver

Download this file:

http://www-user.rhrk.uni-kl.de/~nissler/tuntap/tuntap_tiger.tar.gz

4 Obtain the client software

The software you need is called "gw6c". It is available from the IPv6Now website, www.ipv6now.com.au (click on "Services" then "Client software downloads"), or from the go6 website, www.go6.net (click "Applications", then "Downloads").

Choose the source version on the client – there is no precompiled MAC OS X version.

5 Install the tuntap driver

After obtaining the tuntap driver source code, do the following to install it (note that you must be root for the final step):

```
mkdir tuntap_tiger
cd tuntap_tiger
tar -zxf ../tuntap_tiger.tar.gz
installer -pkg tuntap_installer.mpkg -target /
```

6 Install the client software

You will have to compile the client yourself. Luckily this is generally easy.

Unpack the source package somewhere convenient, **change to the tspc-advanced subdirectory**, and read the INSTALL file there for full instructions.

For MAC OS X, you need to make one specific change to one source file. Edit the file src/lib/md5.c using your favourite text editor, and change this:

```
#include "md5.h"
```

to this:

```
#include <stdint.h>
```

```
#include "md5.h"
```

Now, after making sure that you are in the tspc-advanced subdirectory, do this:

```
chmod +x template/*.sh
make target=darwin clean
make target=darwin all
```

We do recommend you read the INSTALL file. Really, it's a good idea.

Once compiled, install the client wherever you want using this command:

```
make target=darwin installdir=/path/to/install/dir install
```

A good choice of install directory would generally be "/usr/local/gw6c".

Compilation normally requires no special privileges, but installing anywhere except your home directory will usually have to be done as the root user.

If you would rather not install the client as root, you don't have to. You can run the gw6c executable directly out of the tspc-advanced/bin directory.

7   Common install/compile problems

If you see error messages when you compile the client software, double check that you were in the tspc-advanced directory before you started the compile.

Also, double check that you have made the necessary change to md5.c.

If you see errors like these when you try to run the client:

```
2008/07/13 14:09:53 E gw6c: Failed to open tun device: /dev/tun0.
2008/07/13 14:09:53 E gw6c: Failed to initialize tun device.
2008/07/13 14:09:53 I gw6c: Done.
```

.. it usually means you did not install (or did not properly install) the tuntap driver.

8   Configure the software

The client is configured by editing a configuration file. By default, the configuration file is called gw6c.conf and is located in the same directory as the executable. For example, if you installed to /usr/local/gw6c, the executable and the configuration file will both be in the directory /usr/local/gw6c/bin.

Edit the configuration file with your favourite text editor. The file contains lots of useful comments. The software also comes with a Gateway Client Guide in PDF form. This has lots of information about the various configuration options.

After compiling and installing the software, the configuration file will have been created and most settings should already be correct. You should need to change only a few items. However, since different users may have different requirements, this section will describe all of the settings you may need to change. It wouldn't hurt to check that they are all correct for your requirements.

Here are the things that need to be checked and, if necessary, changed:

Set "userid" to the user name supplied to you by IPv6Now. For example, if your user name was "fred":

```
userid=fred
```

Set "passwd" to the password supplied to you by IPv6Now. For example, if your password was "qwertyuiop":

```
passwd=qwertyuiop
```

Set "server" to the name or IP address of the IPv6Now tunnel broker:

```
server=broker.ipv6now.com.au
```

Set "tunnel_mode" to "v6anyv4":

```
tunnel_mode=v6anyv4
```

Set the configuration item "auth_method" to "any":

```
auth_method=any
```

The remaining items are relevant only if you will be requesting a prefix. Note that your account must be one that includes a prefix allocation for this to work. IPv6Now "Try6" accounts do not include a prefix allocation.

Set "host_mode" to "router":

```
host_mode=router
```

Set "prefixlen" to the desired prefix length (normally "64").

```
prefixlen=64
```

Set "if_prefix" to the name of the interface that will do router advertising. Unless you are in more than one network, you would normally choose the interface through which you are connected to the Internet. For example, if the interface that should do router advertising is "eth0":

```
if_prefix=eth0
```

Optionally set "dns_server" to the name(s) of the nameserver(s) that will be handling reverse lookups for your prefix. If you supply any nameserver names, the IPv6Now broker will arrange to delegate your prefix to the nominated nameserver(s). For one nameserver:

```
dns_server=ns1.mydomain.com.au
```

Or use a colon to delimit more than one nameserver name:

```
dns_server=ns1.mydomain.com.au:ns2.mydomain.com.au
```

9   Run the client software

Simply run the gw6c executable. It will be wherever you installed it. Alternatively, you can run it directly out of the tspc-advanced/bin subdirectory, below where you unpacked the source package.

Important: The client must be run as the root user!

If you are using the configuration file in the default location, you can just start gw6c with no arguments and it will use the default configuration file. Otherwise, specify the name of the configuration file on the command line, so the whole command looks something like this:

```
/path/to/gw6c -f /path/to/gw6c.conf
```

In general, it is a good idea to always specify the configuration file in full.

If all goes well, the client will set up a tunnel, and you should now have an IPv6 address (and a prefix if you requested one) and be ready to go!

10   Check the software is working

First check that you have a tunnel interface:

```
ifconfig -a
```

You should see an interface called "tun" or "sit" or similar, with a hardware address of all zeroes. The interface should have an IPv6 address – this is your "client tunnel endpoint" - the address of your end of the tunnel that has just been set up.

You should be able to ping IPv6 addresses on the Internet:

```
ping6 www.kame.net
```

If you have requested a prefix, you should see an address with that prefix on the interface you specified for router advertising.

The client will also have started another process running, a router advertiser daemon. If you have other IPv6-enabled machines in your network, then you should find that they now have addresses with your prefix as well! You could try tracing the route from such a machine to IPv6Now:

```
traceroute6 -n www.ipv6now.com.au
```

That's it – you are on the IPv6 Internet. And, if you requested a prefix, so are all the other IPv6-enabled machines on your network.

11   Stopping the tunnel

Send the running gw6c process a HUP signal. The client will terminate, clearing up its network interfaces and stopping the router advertiser (if any). For example, if the process ID of your running client is 4789:

> kill -HUP 4789

Alternatively, if you have only one tunnel (or more than one but want them all to be stopped) you can avoid having to know the process ID:

> killall -HUP gw6c

It is important to use a HUP signal rather than (for example) TERM or KILL. If you use HUP, the running client will terminate any running router advertiser for you, will remove locally configured IPv6 addresses from the interfaces it has been using, and will generally exit gracefully. If you use TERM or KILL, the running client will just exit without cleaning up after itself.

12   Security

When a tunnel is up, your IPv6 traffic is being tunnelled through any IPv4 firewall or packet filtering devices you might have on your network. Those devices are almost certainly inspecting and checking only your IPv4 traffic; your IPv6 traffic is bypassing them.

> **You should take appropriate steps to protect the computer running the Gateway6 client and any other IPv6-enabled**

computers on your network.

How to do this is beyond the scope of this document, but at a minimum you should ensure that the client host and any other IPv6-capable hosts on your network are running local firewalling or packet-filtering software, and that services you do not use on those machines are switched off.

13   A "gotcha" and how to avoid it

Ipv6Now's tunnel service supports only passdss-3des-1, because it is the best encrypted authentication mechanism supported by the tunnel broker.

There is, however, a small "gotcha" with the passdss-3des-1 authentication method: The first time through, the Gateway6 client software will ask you to confirm that you want to connect to the (hitherto) "unknown" broker. If you are running the client from a script and redirecting the output, you will never see the prompt and the client will wait patiently forever.

The same thing may happen if you set up an icon or similar launcher in your window manager.

The simplest solution is to run the client manually the first time, so you can see the prompt and answer "yes". Note that on multi-user operating systems you will have to do this once for each user that uses the client.

Alternatively, if you are using the source distribution of the client software, you can remove the check. Below the directory you unpacked the source in, edit this file:

```
tspc-advanced/src/tsp/tsp_auth_passdss.c
```

Look for this line:

```
if (!ask(HEX_STR_UNKNOWN_HOST_ADD_KEY, host)) { "
```

Immediately above it is a line "#endif"; some lines below it is a line "#ifdef WIN32".

Move the "#endif" line down to immediately above the #ifdef WIN32", save the file, and recompile (see "How do I install the client for other operating systems?").