Global Information Assurance Certification Paper

IPv6 Transition Security

How IPv6 migration can rip your IPv4/NAT security

By Louis Labelle

October 24th, 2003

Table of Content

Abstract

Imagine starting SimCity with 2^{128} dollars! (That's **3.4×10^{38}**). The number and the possibilities are staggering; the challenge becomes an efficiency issue because of the sheer size.
IPv6 is a bit like that and then you add change management hurdles and the nonchalance of North American ISP's that simply do not have the pressure of their Asian and European counterparts.
IPv6 thus represents many challenges, one of which must be security betterment. Some previous practical have already showed IPv6 has security issues and even recent IETF working groups meeting decided to deprecate previously accepted standards for security issues[1].

And the transition from IPv4, with the *famed* network address translator, will have to go safely for this new protocol to gain its grounds. Since there are plenty of addresses for all the IP-enabled computers in the world, one should expect an onslaught Migration (with a big "M"), when some critical-mass popularity is obtained, either by an unforeseen killer app or market pressure.
Transition techniques are still being tested and developed today. In an effort to bring popular acceptance from consumer markets, some of those protocols even apply to home or small office networks with NAT border devices. This paper compares the risks involved between an established Internet technique, 'Configured Tunnels', and the high appeal technique of the Teredo proposal. We shall also examine what general measures can be taken to adequately address any security risks.

Note: To refrain from having "IPv6" written twice in every sentence, this paper will be understood to state IPv6 when referring to IP addresses, unless specified otherwise in the context.
Also note that IP addresses have been sanitized and changed for private IP addresses:

172.16 represents an IPv4 connection
2001:ABCD:EF01::B99 is the same for an IPv6 address.

Introduction

IPv6 in the real world

Doc Searls is a well-known Linux advocate and respected journalist, he once wrote about measuring a technology's popularity by counting the matches on a Google search. As of October 20th, 2003, over 4,190,000 results could be found on Google for the one term "IPv6".
So IPv6 is coming. The solution is now 9 years old, going on 10. But then again, it took more than ten years for the Internet [with IPv4] to become the mega-cultural revolution we all know it to be now.

Security on the Internet should be, and continue to be, a prevailing aspect to its growth. IPv6 is said to be a "new and improved" protocol; [re]built from the ground up and sibling past experience learnt. The transition from IPv4 to IPv6 is more than likely the biggest acceptance challenge faced by the IETF and everyone involved in Internet regulations.
No matter how easy or transparent the transition will work out to be, it must maintain, if not improve the overall safety of Internet access.
Now going back to Google, on the same date, less than 6350 entries showed up for the combined terms "IPv6 security". This certainly raises an eyebrow of concern for this author.

Security or Obscurity?

For all and any host having access to the Internet, security measures must be taken to protect this host from intrusion and other ill-fated schemes ever present on the great network.

NAT Dominance

The Internet nowadays is dominated by the Network Address Translator and its numerous implementations.

NAT was an immediate solution to the IP address depletion problem. It has worked so well, and in a way so badly, that it transformed the Internet. The Internet is no more an end-to-end host experience; it has literally morphed into a masquerade of IP addresses. A client host accessing a web server is a connection seldom accomplished by two hosts' IP stacks bearing Internet unicast addresses anymore. More than likely, each host is on a private network and the connection to the real IP world is managed by a translator device; be it a simple SOHO Internet Router or a heavy-duty firewall.

The main drive behind this study is the interesting fact that none of this masquerading of IP addresses that we now take for granted will occur in the next generation Internet. The facts show that the IPng standard has been elaborated with the objective of eliminating the need for NAT devices and NAT-ed networks.

So, how ready are we to let go of our NAT principle where the IP address of our PC does not exist on the Internet? Where our existence is hidden from the real world?

IPv6 will be the bringer of change by forcing everyone to use a global unicast IP address. This adds a strange sense of vulnerability: as if NAT was a protective shield and IPv6 will be removing it, leaving us bare and exposed.

It is this sense of security that some advocate as being false. NAT is often referred to as an “obscurity” feature rather than a security one.

Having an IPv6 address on the Ethernet interface of that PC now exposes the computer directly to the outside world and this needs to be properly addressed.

No NAT for IPv6

IPv6 did not exactly emulate its sibling protocol in reserving specific address blocks within the scheme for non-routable or translation purposes. Instead, addresses are divided into ranges called prefixes and two such ranges have been specifically reserved for peer IP applications:

- There is a peer-to-peer networking prefix called link-local (FE80::). It is similar to the 169.254/16 network. Since these addresses are truly non-routable, the use of NAT is not possible here at the moment.
- There is a second prefix reserved for private site addressing called site-local (FEC0::), with the capability of subnetting within an organization’s routed private network,

The site-local prefix specification has been deprecated in 2003. One of the main reasons was an apparent leakage of addresses to the Internet during tests and this raised some security concerns[1].

In general, all host interfaces will bind one of these local-only IP prefixes in conjunction with an auto-deterministic protocol to properly assign themselves a unique IP address. Bindings will also apply for the Internet public address, or global unicast address, which will be directly linked to the host’s interface.

If this host is located on a home network, security is not as prevalent as with IPv4. And while some might say NAT is no longer needed, the fact remains this

host is now directly accessible from the Internet and thus, the relative safety of NAT may no longer be applicable for such an IPv6-enabled host.

This subject is very much up to date, since an Internet draft by M. Huitema et al. related among other things this security issue about the Teredo transition protocol[2].

Preliminary ground work

Configured Tunnels Overview

In the configured tunnel's approach, the border device has a dual layered IP stack. It has an ISP assigned IPv4 address, fixed or dynamic and a global unicast IPv6 address used for routing purposes with a small subnet of a few bits for the tunnel connectivity (/127 in our example). On the internal network, the IPv6 addresses can be an address range of 48bits (or /48), and in some cases tunnel brokers will provide a /64. These addresses are part of the allocation done by the tunnel broker.

Configured Tunnel refers to the fact there is human interaction in obtaining a tunnel assignment between client and provider. Once the configuration elements are established, the tunnel can then function on its own with IP as the transport and protocol 41 is used to encapsulate the next generation traffic.

A variant technique is used by the tunnel pioneer Freenet6, owned and operated by Viagenie. It uses the **T**unnel **S**etup **P**rotocol, as per the Internet Draft by Marc Blanchet[3], to establish connectivity. In this technique, the tunnel is set up using a protocol similar to POP and SMTP, with commands and responses, to create a temporary or permanent association between the client border device and the Freenet6 tunnel server. A temporary association will provide a single IP address (/128 mask) and a permanent association could provide a range of 48bit addresses.

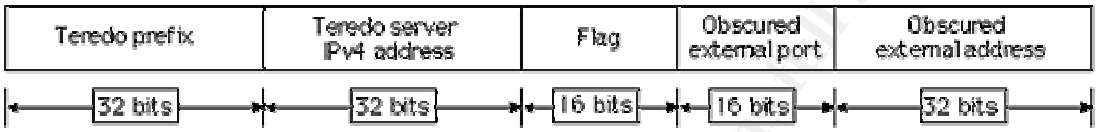
There is a way to make TSP work through some NAT devices[4] but it is very complex to set up and only available to truly flexible firewalls, such as open source Linux or BSD and high-end solutions. This is unfeasible on consumer market appliances.

Note that in this system, border devices use only valid Internet unicast addresses; there are no RFC1918 private IP or any equivalents.

Teredo Overview

Teredo will be using a 32 bit prefix to be determined by the IANA. Meanwhile, the Microsoft implementation of Teredo in the Advanced Networking Pack of Windows XP – SP1 is using Microsoft's '6Bone' assigned prefix of 3FFE:831F::/32.

The 128bit address of a Teredo host on the Internet would appear as such[5]:



Teredo address

The "Obscured external port" and the "Obscured External address" fields are in fact the host's local half of the UDP socket pair that we will see later; the external IPv4 address, e.g. 9.0.0.1 (0900 0001) and the port opened for the communication, e.g. ephemeral port 4096 (1000).

This elaborate and clever addressing scheme allows the protocol to enable many types of direct communication flows from the NAT-ed host to the outside IPv6 world and back. All this is done without any major modifications to the NAT device.

Understandably, Teredo is being designed for the very specific purpose of providing a last resort[6] option for networks with consumer market appliances that would otherwise block 'IPv6 over IPv4' traffic.

This could very well be the *pharmacon*[7] of IPv6 transition meaning both the remedy and the poison. It's a measure that might ease the transition nightmare but at the same time, if it is used in an ill-fashion, it could create a disaster.

How does Teredo elude NAT devices

The Teredo protocol allows a host configured with a private IP address accessing the Internet through a NAT device to communicate with the v6 Internet, using UDP to encapsulate the IPv6 packets. UDP being a connectionless protocol, it is practically impossible for a NAT device to perform any filtering on the packets. This in turn, allows bi-directional traffic flowing through the NAT device, beyond its control. Note that Teredo works on most NAT devices, not all.

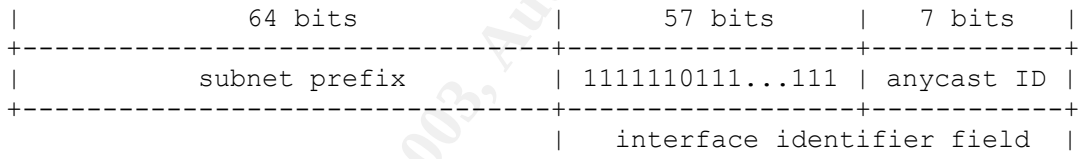
The Teredo host will use the Teredo server to translate its packets to the IPv6 Internet. Teredo servers are not involved in the packets' return path, Teredo relay routers perform this operation. This ensures performance and efficiency by removing the dependency toward a single server or router. And if the server or router goes down in the middle of a transaction, another one can seamlessly pick up where the first one left off.

The main enablers in this technique are the use of an anycast address for the Teredo servers and the specific 32 bit prefix that pinpoints to the group of Teredo relay routers for the return path.

Unicast, Multicast and Anycast

Anycast addresses are not in a range different from the unicast addresses since they must be addressable by the Internet at large.

They are addresses within a network's subnet with the first 64 bits being the network ID, and the remaining bits are the node's address in the subnet; it is divided as such in the RFC2526[8]:



This means there are 7 bits to define an Anycast address, leaving us with no more than 128 values possible. Values are assigned by the IANA organization. The only valid ID assigned thus far is for Mobile IPv6 whose value is set at 126 (0x7E).

Getting connected with IPv6 today

Teredo service

Teredo is still an Internet Draft.

Christian Huitema, a Microsoft architect, is actively involved in the continued effort to develop the protocol. Strong in its market position, Microsoft has forged ahead and implemented Teredo in its Best-of-bread products either as a full-fledge feature of Windows 2003 Server or as an add-on in Windows XP.

Configured Tunnels

Hurricane Electric is offering a tunnel broker service, free as of this writing.

Freenet6, from Viagénie, is another tunnel broker service, with its own **T**unnel **S**etup **P**rotocol.

Transition techniques

Teredo

Definition

Teredo holds its name from the mussels called *Teredo Navalis***,** or shipworm. Shipworm was the original name of the Teredo technique but fear of confusion or simply the negative association to Internet worms made the protagonists change its name.

The analogy is to the fact these mussels would attach to ship hulls and travel the open seas at the expense of the carrier. It is very interesting to read about the original shipworm, since it was actually a pest, causing damage to its hosts[9].

How does it work

The Teredo proposal[6] calls for many players, here is brief listing:

Teredo Server	Dual layer IP, responding to IPv4 anycast Teredo address
Teredo Relay	Dual layer IP router in the Teredo Prefix, forwards packets
Teredo Host	Private IP address, 10.0.0.5
Teredo 2nd Host	Private IP address, 192.168.0.5
IPv6 host	IPv6 only host
NAT device	9.0.0.1 on the Internet and 10.0.0.1 on the private network.

Here is a sample communication between Teredo Host and IPv6 Host:

- The host opens up a UDP connection to the Teredo server, using its Teredo anycast address. Within this packet is the real destination of the IPv6 connection. The socket pair (host inside:port -> Teredo server:port) will be remembered during a certain time by the NAT device. By sending regular meaningless packets, called bubbles, to the outside world, the host ensures this “hole” is kept open and functional.

- The nearest Teredo server listening to the anycast address receives the packet and then converts it and sends it to the v6 Internet, to the real destination. The IP address used to send the packet is constituted of the concatenation of the Teredo prefix, the Teredo anycast IPv4 address, some flags and the obscured address and port of the “Teredo” host.

- The destination host receives the packet and responds to the constructed address. Since this address begins with the Teredo prefix, it is routed to the Teredo subnet.

- An active Teredo relay then receives the packet, extracts the IPv4 destination and Teredo anycast addresses and forwards it to the NAT device for translation back to the “Teredo” host.

  The packet leaving the Teredo Relay will bear as source the IPv4 anycast address of the Teredo server, so this way the NAT device will accept the communication.

  Note that any Teredo relay can perform this task, the initial Teredo server is not involved.

Another sample communication, this time between two Teredo Hosts:

- The first host obtains knowledge of the second host’s IPv6 address by means of DDNS or some directory service. It also learns the outside IPv4 address and port of the second host by extracting it from the IPv6 address

- The first host sends the first packet to the Teredo server, using the Teredo anycast address. Within this packet is the IPv6 address of the second host.

- While the first host sends the UDP packet to the Teredo server, it also sends a UDP bubble directly to the second host. This packet will be rejected by the NAT device of the second host since there are no previous connections.

- The nearest Teredo server listening to the anycast address receives the packet and extracts the second host IPv4 address, encapsulates the packet in UDP and sends it to the second host. The IP address used to send the packet the Teredo anycast address.

- The NAT device of the second host receives the packet and translates it to the second host, since the socket pair with the Teredo anycast address is always opened.

- In its response to the first host, the second host begins the same process, sending to the Teredo server and a bubble directly to the first host.

- Upon traversing the second host's NAT device, the bubble will create the needed inbound socket pair mapping for packets coming from the first hosts' outside address. Upon reaching the first host's NAT device, it will be accepted since the first host opened up the socket pair when it initially sent the bubble to the second host's outside address.

- When the first host receives the bubble coming from the second host, it knows it has direct contact and the Teredo server is no longer needed.

- Both hosts will now communicate directly to one another in IPv6 over UDP and their respective NAT devices are defenseless in this exchange.

Platform specific issues

Teredo not being officially implemented, it is too early to obtain such information.

Implementation issues

There is a listing of supported NAT devices on Microsoft's web site[10]

Tunnel Broker services

What are the different types

Tunnel brokers use simple routing commands on the host or the router to accomplish the connectivity. Mostly, the transition is accomplished by the IP stack which is then composed of one or two pseudo-interfaces, depending on the platform.

The routing command will basically announce that all IPv6 routes, or ::/0 (the IPv6 equivalent of 0.0.0.0) is reachable through the tunnel server's address.

What is important to understand behind all this is that there is no encryption. This is not an L2TP or IPSec tunnel, although it bears some resemblance.

The tunnel is simply a routing encapsulation performed by the multiple interfaces declared in the IP stack

How does each one work

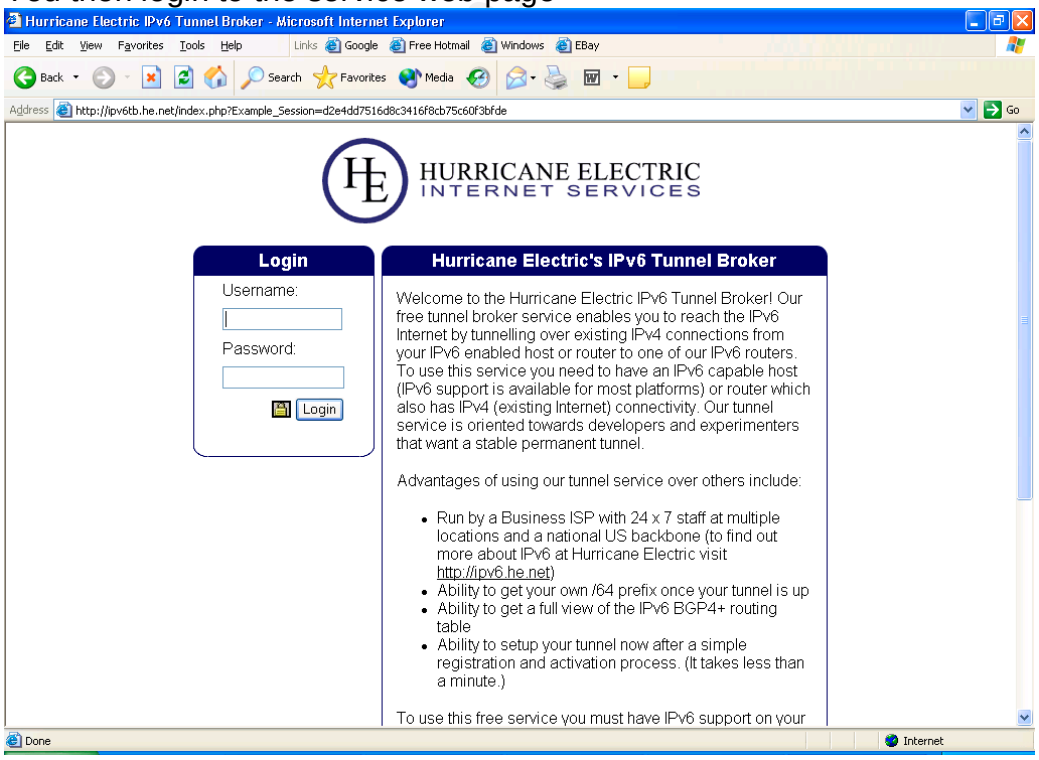
Listing of all Tunnel Brokers and their configurations is beyond the scope of this paper, so only two providers are examined; Hurricane Electric and Freenet6.

Freenet6 is somewhat particular since it uses the [still drafted] TSP client to establish the connection. For anonymous connections, there is no web interface interaction unless one registers and opens an account to obtain an IP range. Simply download the client and edit the tspc.conf file accordingly. Freenet6 does support Cisco routers, the client is run externally and commands are then sent to the router.

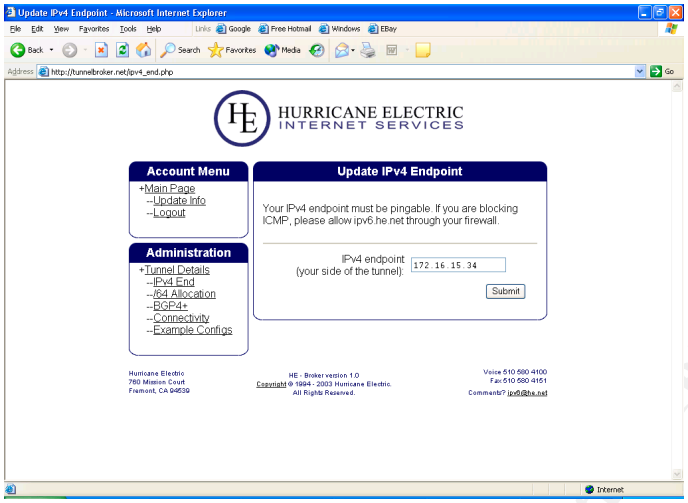
Hurricane Electric functions with simple registration through a login name, and then an IP address block is assigned. The next slides will show you how the human intervention is done in a typical Configured Tunnel approach.

First, you must complete a registration process, similar to opening an e-mail account. This creates a tunnel assignment with a username and password of your choice.

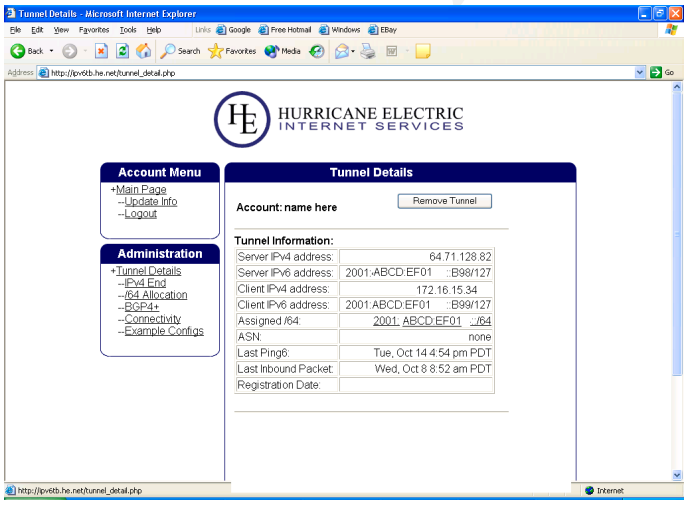
You then login to the service web page



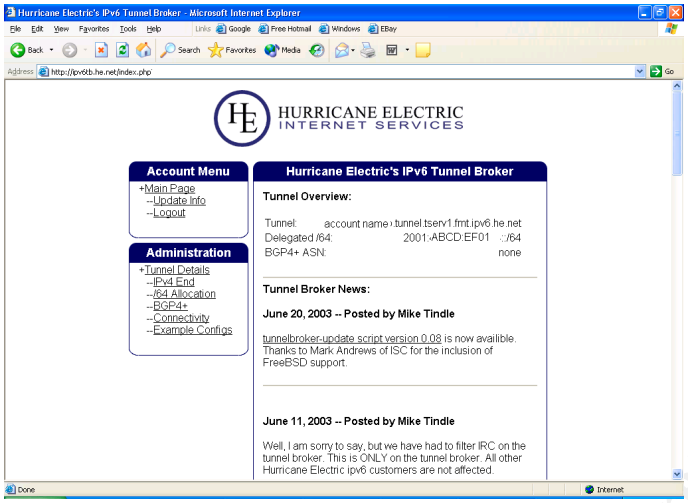
In the menu on the right, you select IPv4 End to enter your ISP provided current IP address. When clicking Submit, the server will try to ping your endpoint address, so make sure you accept ICMP echo-request from their server, 64.71.128.82



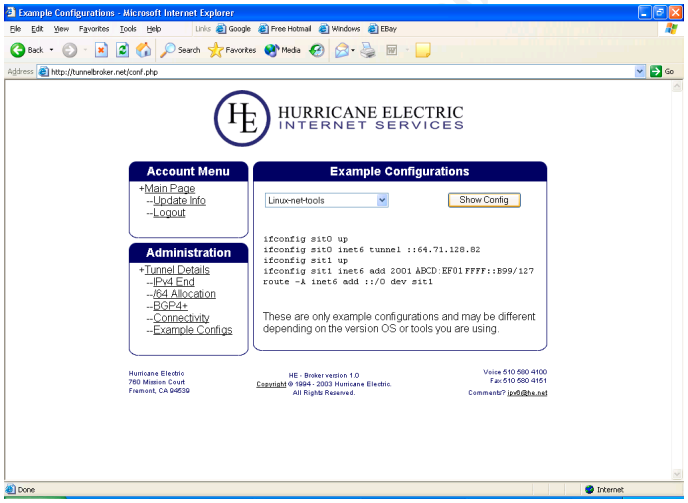
In the Tunnel Details, you will see the dual layer approach, as each IP address is shown. The Server side IP addresses correspond to the commands that HE will suggest you use in configuring your host or router device. The client IP addresses represent the assignment you have been awarded.



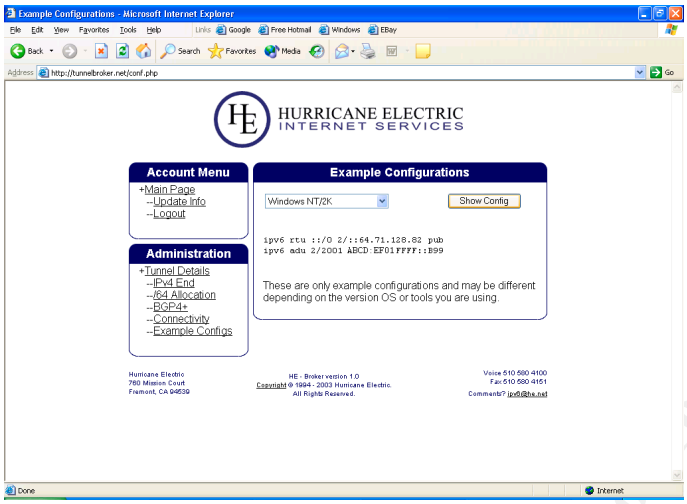
The Tunnel Overview shows your DNS AAAA record and its assigned IP address.



The Example Configurations menu is a suggestion of commands needed to add and configure the interfaces on your host, or router. Hurricane Electric lists a few typical configurations; here we can see the Linux and also the Windows configuration on the next slide.



This Windows configuration is obsolete because of the commands used, Microsoft now recommends using the netsh command line interface for all IPv6 configurations.



There is a page on Microsoft's Windows 2003 Server web site[11] that can help you translate ipv6.exe commands to their netsh.exe counterpart.

For our example, here is the translation:

From:

ipv6 rtu ::/0 2/::64.71.128.82 pub (To declare the IPv6 default gateway)

ipv6 adu 2/2001:ABCD:EF01:FFFF::B99 (To declare the IPv6 address on the local interface)

To:

netsh interface ipv6 add route prefix=::/0 int=2 nexthop=::64.71.128.82 pub=yes
netsh interface ipv6 add address int=2 addr=2001:ABCD:EF01:FFFF::B99

Platform specific issues

Here are the pseudo-interfaces created for Linux and Windows

Linux:
Sit0: Used to configure the IPv4 endpoint, providing encapsulation on IPv4, proto 41
Sit1: Used to configure the endpoint with its global unicast address

Windows:

'Tunnel adapter Automatic Tunneling Pseudo-Interface': Used to configure the endpoint with its global unicast address

Implementation issues

Most recent operating systems versions already include IPv6 functionality, or add-ons can be found.

The nice aspect of Configured Tunnels is the total absence of proprietary software for any platform. It is basically a routing system. This is all that is required as the reader will have just seen. The TSP client of Freenet6 still falls into this statement since the protocol and the Freenet6 client are only used to establish the tunnel and configure appropriately the routing on the host. TSP is not involved in the actual routing or sending of packets.

This approach is best suited for a managed border; whether using namesake firewalls or Linux / BSD solutions; one can activate next generation Internet access and obtain complete site connectivity without impeding on security standards.

This being said, it is beyond the scope of this effort to document all the changes required by leaving the IPv4 and NAT realm of security and migrating to the new IP standard. Documents have been written about running an IPv6 Firewall, including practical by SANS' certified graduates. The author recommends reading C. Schneeberger's GSEC practical of January 2003, labeled "Firewalling IPv6 with OpenBSD's pf".

Security risk of Teredo

As the reader may have noticed, there are possibilities to forge addresses within the Teredo technique, most notably in the anycast address used by the host behind the NAT. The following netsh command will even help you do this :

netsh interface ipv6 set teredo servername=w.x.y.z

This could enable someone with malicious intent to send all IPv6 packets over UDP to its own Internet server, where packets would be accepted. This behavior is very similar to a successful intrusion.

Unsuspecting users of updated Windows XP computers could have IPv6 enabled without their knowledge by the simple add-on of Microsoft's Advanced Networking Pack. The NAT device or firewall, thought to be protecting the network, would be rendered defenseless and open communication could result.

The concept of piercing through NAT devices by encapsulating over UDP is not new, it is a well known technique used by gaming consoles, such as the XBox or PS2.

There are many types of NAT devices; Cone, Restricted Cone, Port-Restricted Cone and Symetric. Of these types, only the symetric NAT is not supported by Teredo. For a detailed explanation of NAT types, see this presentation[12] by Takuya Oikawa of Microsoft published on Ipv6style.jp, a Japanese web site dedicated to IPv6.

The security issue is in the integration of this technique with the most popular computer OS… That should raise a chill to anyone concerned with security.

How long before a worm or virus is developed using this ability and creating yet another Internet hell?

Securing against Teredo

One way to protect a network against unwanted Teredo traffic will be to monitor the IANA publications for the standard UDP port selection for the proposal and to block all outbound access through that port.

But for the sake of argument, what if someone changes the UDP port used to DNS' UDP port 53? Then only stateful inspection of UDP known ports can help block the misuse of open ports.

Again, NAT devices are defenseless in this type of action unless manufacturers change the features in the firmware. Teredo is at our doorstep, especially with the Microsoft Network Pack made available through simple download, so there might be a delay between market reaction and the playing field.

Host-based protection

Microsoft netsh command

As previously mentioned, Microsoft created a new tool for network configurations. It is the executable netsh.exe.

There is information available on the Windows XP Home Edition site[13] to understand NETSH and the added-value it brings. Netsh is available on Windows 2000, Windows 2003 Server and Windows XP.

NETSH is also used to enable the IPv6 stateful firewall[14] through command line interface. The GUI interface is only valid for Windows XP and the IPv4 connections.

In order to use Teredo to connect to the v6 Internet, one should carefully read the netsh help and documentation to create a series of commands that can be applied to hosts intended to use the Teredo service.

In general, netsh is both a command line configuration and inspection utility and environment. Through a single command line, one can set or display settings. At the same time, by starting the netsh itself, one can navigate in the tree of menu and commands to perform many operations.

An exhaustive study of NETSH is beyond the scope of this paper, but for the sake of giving an example, here is the command to make sure ICMP echo-request packets are currently dropped:

netsh firewall show adapter “Teredo Tunneling Pseudo-Interface”

And the result should resemble this:

```
Description                                 ICMPTypeNo  Enabled
-----------------------------------------------------------------
Allow Outbound Destination Unreachable      1           No
Allow Outbound Packet Too Big               2           No
Allow Outbound Time Exceeded                3           No
Allow Outbound Parameter Problem            4           No
Allow Inbound Echo Request                  128         No
Allow Redirect                              137         No
```

Transition outlook

Home network

It is way too early to forecast how the consumer market will react to the advent of IPv6 transition. NAT devices are being sold at very attractive prices and some may be upgradeable. With north-American ISP stalling the growth of IPv6, and broadband Internet booming the way it is, change is not bound to happen too soon.

There should be hope to have a better solution for the blooming home networks through consumer products, and in the mean time, European and Asian ISP will be leading the way. May they find a better solution than Teredo and share the experience by the time our ISP's are ready.

Corporation network

There are numerous efforts documenting the different strategies known to address the challenges of IPv6 transition.

As we have seen, keeping current security standards at a company is better accomplished in using configured tunnels with a tunnel broker. If ISP's start offering tunneling as a feature, then maybe companies can include this in their RFP selection criteria.

In the meantime, companies should have strict policies about UDP traversing firewalls and make sure they put together an IPv6 policy statement somewhere in the coming years. This will help position IPv6 and open the way for a carefully planned transition strategy in due time.

References

Hagen, Silvia. IPv6 Essentials.
Oreilly, July 2002.

1 Huitema, Christian. “Deprecating Site Local Addresses.” IETF working group meetings”, version 01, October 13th, 2003
URL: http://www.ietf.org/internet-drafts/draft-ietf-ipv6-deprecate-site-local-01.txt
(October 20th, 2003)

2 Huitema, Christian. "Evaluation of Transition Mechanisms for Unmanaged Networks", 26-Jun-03.
URL: http://www.ietf.org/internet-drafts/draft-ietf-v6ops-unmaneval-00.txt
(October 07, 2003)

3 Blanchet, Marc “Tunnel Setup Protocol (TSP): A Control Protocol to Setup IPv6 or IPv4 Tunnels”, version 01, July 1, 2002
URL: http://www.freenet6.net/draft-tsp.shtml
(September 30th, 2003)

4 O'Gorman , Jim. “IPv6 Behind a NAT”, September 2000
URL: http://www.daemonnews.org/200009/ipv6.html
(September 30, 2003)

5 Oikawa , Takuya. “Advanced Network Pack for Windows explained.” Trying Out for Yourself. September 29th, 2003
URL: http://ipv6style.m-t.com/en/tryout/20030929/2.shtml
(October 1st, 2003)

6 Huitema, Christian. “Teredo: Tunneling IPv6 over UDP through NATs.” Internet Draft v6ops. June 6, 2003
URL: http://www.ietf.org/internet-drafts/draft-huitema-v6ops-teredo-00.txt
(October 8th, 2003)

7 HyperDictionnary. Medical dictionary. Ref. Webster 1913
URL: http://www.hyperdictionary.com/dictionary/pharmacon
(October 10th, 2003)

8 Johnson D. and Deering S. RFC2526 "Reserved IPv6 Subnet Anycast Addresses." March 1999
URL: ftp://ftp.rfc-editor.org/in-notes/rfc2526.txt

9 Ruiz, Bruce C.. "Teredo Navalis." Panama History. January 18th, 2003
URL: http://www.bruce.ruiz.net/PanamaHistory/teredo_navalis.htm
(October 15th, 2003)

10 Microsoft Corporation. "Windows Peer-to-Peer Networking." Windows XP Home Page. September 25, 2003
http://www.microsoft.com/WindowsXP/p2p/natcompat.asp
(October 21st, 2003)

11 Microsoft Corporation. "Updating IPv6.exe Commands to Netsh Commands." Windows 2003 Server - Technology Centers. August 15, 2002
URL:http://www.microsoft.com/windowsserver2003/technologies/ipv6/ipv62netshtable.mspx
(September 25th, 2003)

12 Oikawa , Takuya. "Advanced Network Pack for Windows explained." Trying Out for Yourself. September 29th, 2003
URL: http://www.ipv6style.jp/en/tryout/20030929
(October 1st, 2003)

13 Microsoft Corporation. "Using NETSH." Windows XP Home Edition Product Documentation.
http://www.microsoft.com/windowsxp/home/using/productdoc/en/netsh.asp
(September 25th, 2003)

14 Microsoft Corporation. “To configure IPv6 Internet Connection Firewall.” Technet - To configure IPv6 Internet Connection Firewall. http://www.microsoft.com/technet/itsolutions/network/maintain/security/ipv6fw/hcfgv601.asp?frame=true (September 25th, 2003)