HIP experimentation using Teredo

Robert Moskowitz
ICSAlabs
an Independent Div of
Verizon Business Systems

July 28, 2008

rgm@labs.htt-consult.com

Project Goal

- Total commitment to IPv6

- Forget IPv4 and NATs

  - Well, almost as we shall see

- Serious IPv6 occuring over next 12 months

  - US gov now a main driver

  - But so is the gaming world

What is Teredo?

- ## RFC 4380

  - Tunneling IPv6 over UDP through Network Address Translations (NATs)

- ## Teredo navalis

  - Is the latin name of one of the best known species of shipworm

  - The idea was that the protocol would pierce holes through NAT
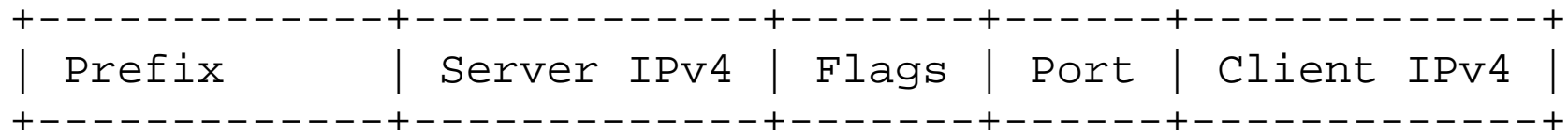
So why add one more Indirection?

- RFC 1925 Rule 6
  - It is easier to move a problem around (for example, by moving the problem to a different part of the overall network architecture) than it is to solve it.
- Teredo has been shown to work for the gaming community
  - Built into Vista and XP
- Most NATs types work with it
  - Along with multi NAT traversal

Teredo Architecture

- ## Teredo Server

  - A node that has access to the IPv4 Internet through a globally routable address, and is used as a helper to provide IPv6 to Teredo clients.

- ## Teredo Relay

  - An IPv6 router that can receive traffic destined to Teredo clients and forward it using the Teredo service.

- ## Teredo Client

  - A node that has some access to the IPv4 Internet and wants to gain access to the IPv6 Internet.

What does Teredo supply?

- An IPv6 address:

```
+-------------+-------------+-------+------+-------------+
| Prefix      | Server IPv4 | Flags | Port | Client IPv4 |
+-------------+-------------+-------+------+-------------+
```

  - Prefix: the 32-bit Teredo service prefix

  - Server IPv4: the IPv4 address of a Teredo server

  - Flags: a set of 16 bits that document type of address and NAT

  - Port: the obfuscated "mapped UDP port" of the Teredo service at the client

  - Client IPv4: the obfuscated "mapped IPv4 address" of the client

Testing Environment

- Native IPv6 and public IPv4 allocations available

- Teredo Server and Relay (on single system).

- 2 IPv4 NAT gateways

  - One supporting 'hairpin' routing, other not

- IPv6 SIP PBX

  - e.g. Asteriskv6

  - With HIP

- IPv6 SIP client(s) (minimum of 2)

  - Native, Tunneled, and Teredo IPv6

Test Plan

- HIP working with IPv6 native
- SIP over HIP
  - Peer to Peer
  - Peer to PBX to Peer
- HIP with Teredo
  - All Teredo scenarios
  - Including relaying
- Mobility Testing
  - Move those clients
  - And PBX!

RFC 1925 Rule 8

- It is more complicated than you think

- Mobility means

  - Get new IPv4 address

  - Get new IPv6 address

    - If no native or tunneled IPv6 found

    - Get new Teredo IPv6 address

  - Perform HIP rendezvous

  - Continue SIP session with no changes

- So there is enough traffic to interrupt the call

Status

- Factors attributed to 2+ month project delay

- Native IPv6 working 2+ weeks

- Teredo (Miredo) server and gateway working 1 week

- HIPL in userspace on Centos 5.2 working 4 days

- No SIP client selected

Observations

- Still believe this will work

  - The gaming community went IPv6 via Teredo, HIP/SIP can too.

- But mobility will not be magically smooth

  - Teredo cost may be substantial

    - Need some real numbers on packets and timings

  - Feel that Teredo/native IPv6 beats NICE/ICE in all ways

- What are 'mainstream Unix platforms that need HIP?

  - I use Centos and HIPL for FC8 did not work

Next up

- I brought enough gear here to do make progress this week

- Will 'walk the talk' after I get tunneling set up and switch from SSH tunneling to HIP for everyday use on Centos

  – Need this for SIP anyway!

Questions?