Confidence in a connected world.

Vista's Network Attack Surface

Dr. James Hoagland, Principal Security Researcher

Work with Ollie Whitehouse, Tim Newsham, Matt Conover, Oliver Friedrichs

Symantec Security Response – Advanced Threat Research

CanSecWest, April 2007

Windows Vista Network Attack Surface Analysis

- Symantec Advanced Threat Research (ATR) conducted a project looking at “network attack surface” of Vista
- We examined the security-relevant aspects of Vista, from the point of view of the network
- Huge potential scope, actual scope was defined by time available
- Very broad review, from layer 2 to 7
- Focused on the default (out-of-the-box) configuration
- We were able to dig fairly deep into some areas

Agenda

1 Introduction

2 Windows Vista Firewall

3 Layer 3

4 Layer 4

5 Teredo

6 Additional results

Symantec ATR Vista Research

This network attack surface analysis is part of Symantec ATR’s batch of Vista research, conducted in conjunction with its initial public release

- Lots of systems will be running Vista so it's important to know what to expect
- We began our study with beta builds
- RTM (release) results are available at:
  - http://symantec.com/enterprise/theme.jsp?themeid=vista_research (http://tinyurl.com/ynr2b8/)
  - Almost all results presented here are from this build

We've produced several public research papers on Vista and Vista-related technologies…

Symantec ATR Vista Reports (1)

Last July-August (Vista Beta 2 builds):

- *Windows Vista Network Attack Surface Analysis: A Broad Overview*
  - By Tim Newsham and Jim Hoagland
  - http://www.symantec.com/avcenter/reference/ATR-VistaAttackSurface.pdf
- *Analysis of the Windows Vista Security Model*
  - By Matt Conover
  - http://www.symantec.com/avcenter/reference/Windows_Vista_Security_Model_Analysis.pdf
- *Assessment of Windows Vista Kernel-Mode Security*
  - By Matt Conover
  - http://www.symantec.com/avcenter/reference/Windows_Vista_Kernel_Mode_Security.pdf

Last November:

- *The Teredo Protocol: Tunneling Past Network Security and Other Security Implications*
  - By Jim Hoagland
  - http://www.symantec.com/avcenter/reference/Teredo_Security.pdf

Symantec ATR Vista Reports (2)

Last February (RTM build):

- *Security Implications of Windows Vista*
  - By Oliver Friedrichs and Ollie Whitehouse
  - http://www.symantec.com/avcenter/reference/Security_Implications_of_Windows_Vista.pdf
- *The Impact of Malicious Code on Windows Vista*
  - By Orlando Padilla
  - http://www.symantec.com/avcenter/reference/Impact_of_Malicious_Code_on_Vista.pdf
- *Analysis of GS Protections in Windows Vista*
  - By Ollie Whitehouse
  - http://www.symantec.com/avcenter/reference/GS_Protections_in_Vista.pdf
- *An Analysis of Address Space Layout Randomization on Windows Vista*
  - By Ollie Whitehouse
  - http://www.symantec.com/avcenter/reference/Address_Space_Layout_Randomization.pdf

Symantec ATR Vista Reports (3)

Last March:

- *Windows Vista Network Attack Surface Analysis*
  - By Jim Hoagland, Matt Conover, Tim Newsham, Ollie Whitehouse
  - http://www.symantec.com/avcenter/reference/Vista_Network_Attack_Surface_RTM.pdf
  - Covers RTM (release) build of Vista
    - Complete retest
    - Deeper dive into parts
  - This paper is the main basis for this presentation
    - Only have time for highlights of results, see the paper for details
- *The Teredo Protocol: Tunneling Past Network Security and Other Security Implications* (updated version)
  - By Jim Hoagland
  - Platform-independent assessment
  - http://www.symantec.com/avcenter/reference/Teredo_Security.pdf
  - Another focus of this presentation

The Vista Network Stack

- The TCP/IP stack was rewritten in a major way for Vista
  - Many behavior changes
  - Nmap fingerprint quite different
  - Provides something interesting to study

- New stack = more vulnerabilities?
- Rewritten stack means lots of opportunity for vulnerabilities
  - 1000's of lines of new code
  - Stacks are complex entities that takes years to mature
  - Though we're sure that the extensive testing and security design process that Microsoft has been doing has eliminated many possible vulnerabilities
- There could also be more attacker focus on the new stack since they expect there to be more bugs

Vista and IPv6

- Microsoft loves IPv6
  - “Microsoft’s Objectives for IPv6” (http://www.microsoft.com/technet/network/ipv6/ipv6.mspx)
  - Global addresses and the absence of NAT means peer-to-peer and games are easier to set up
    - More attacker opportunity though too
- On Vista, IPv6 is enabled and preferred by default
- Integrated IPv6/IPv4 stack
- Stack provides IPv6 transition mechanisms such as Teredo

New Protocols in Vista

New protocols include:

- IPv6-related
  - IPv6 (plus six extension headers)
  - ICMPv6
  - NDP (Neighbor Discovery Protocol)
  - MLDv2 (Multicast Listener Discovery)
  - Teredo
  - ISATAP
- LLTD (Link Local Topology Discovery)
- LLMNR (Link-Local Multicast Name Resolution)
- SMB2
- PNRP (Peer Name Resolution Protocol)
- PNM (People Near Me)
- WSD (Web Services on Devices)

A number of other protocols were reimplemented as well

- IPv4, TCP, UDP, ICMP, ARP, IGMP, etc

Agenda

1 Introduction

2 Windows Vista Firewall

3 Layer 3

4 Layer 4

5 Teredo

6 Additional results

Windows Firewall Rules

There is a new version of Windows Firewall for Vista

- Windows Firewall has sets of rules (exceptions) organized into groups
- Rules are often enabled/disabled by group
- Rule can be bound to specific protocol, local port, remote port, local address, remote address, and/or program
- Vista introduces network profiles
  - Each network interface is in one profile at a time
  - 3 built-in network profiles
    - Public (default)
    - Private (home or office)
    - Domain (under a domain controller)
  - Vista automatically assign profiles, with user input
  - Each firewall rule can be in one or more of the profiles
  - Thus the network profile selects a firewall ruleset

We focused on inbound firewall filtering

Windows Firewall Initial Status

- Firewall is on by default (good)
- Limited exceptions by default
  - Core Networking group (all profiles)
  - Network Discovery group (private profile)
  - Remote Assistance group (private profile)
- All TCP and UDP rules in initial ruleset without a specific port are at least bound to a specific program

Windows Firewall State Change Testing

- We wanted to study the effect of GUI actions on Vista on Windows Firewall and active sockets
  - E.g., enabling file sharing, turning back off
- Enabling certain features opens Windows Firewall exceptions (after consent prompts)
  - However, we observed that these exceptions don't always go away when the feature is disabled
  - Leftover exceptions even persist across a reboot
  - Thus a legacy of firewall exceptions builds up until manually disabled

Windows Firewall Sticky Rules

In our limited study we noticed:

- Turn Media Sharing on then off:
  - "Windows Media Player" group remained enabled (private and domain profiles)
- Sign into People Near Me then quit it:
  - "Windows Peer to Peer Collaboration Foundation" group remained enabled (all profiles)
- Sign into Windows Meeting Space then quit it:
  - "Windows Peer to Peer Collaboration Foundation", "Windows Meeting Space", and "Network Projector" groups remained enabled (all profiles)

Of course, need a listener + a firewall exception for a port to be open

- Sockets usually closely matched service state
  - Though TCP port 5722 (DFSR.exe) remained open an extra few minutes after Windows Meeting Space
- Sticky rules still increase exposure though

Agenda

1 Introduction

2 Windows Vista Firewall

3 Layer 3

4 Layer 4

5 Teredo

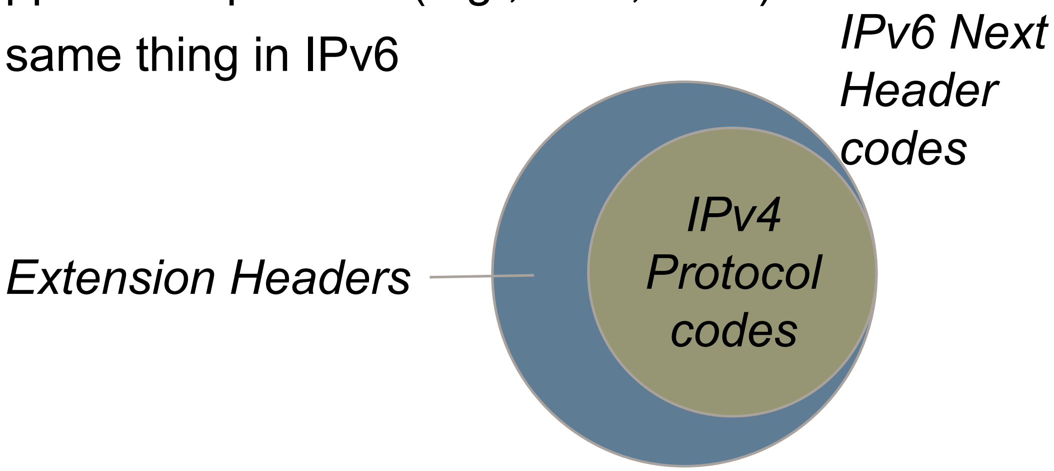
6 Additional results

IPv6 Security Implications

- Vista is the first time IPv6 is enabled by default in Windows
- IPv6 has many implications for security that we have previously studied
  - (sorry, we haven't published on this as yet)
- The following security implications apply in general to IPv6 implementations/installations:
  - A network's security controls may not be ready for IPv6
    - Or may not be configured properly (e.g. not applying a firewall rule to IPv6 as well as IPv4)
  - New (less tested) code would be present in the stack and applications
  - IPsec is a standard part of IPv6, providing encryption and authentication
    - But there are challenges to actual use
  - Blind scanning of Internet addresses is infeasible generally
    - Though there are still other methods of host discovery
  - Tunneling raises security concerns
  - Easy local NDP attacks
    - Pretend to be a router, DOS a host or network, etc
  - Link-local addresses can prove host locality
  - And many more

Background: IPv6 Next Header vs. IPv4 Protocol Fields

- The “Next Header” field in IPv6 is much like the “Protocol” field in IPv4
  - Used to indicate upper level protocol (e.g., TCP, UDP)
  - IPv4 codes mean same thing in IPv6



- However, the IPv6 Next Header is also used for IPv6 extension headers
  - A Next Header field is also located in extension headers, to indicate what follows
    - So, there can be multiple extension headers
  - Extension headers include: Dest Options, Hop-by-hop Options, AH, ESP, Fragment, Routing, Mobile IPv6

IPv4 Protocol/IPv6 Next Header Enumeration

Protocols/codes	IPv4	IPv6
Unsupported protocol codes	No response with firewall on – tested with it off	Produced a param prob msg, so we can map serviced protos
ICMPv4	Yes	
ICMPv6		Yes
IPv4 over IPv_	**Yes**	**Only if firewall on**
IPv6 over IPv_	Yes	**Yes**
GRE	Yes	
IGMP	Yes	
TCP & UDP	Yes	Yes
ESP & AH	Yes	Yes
Routing/43 & Fragment/44	**Yes**	Yes
Hop-by-hop & Dest. Opts		Yes
IPv6 No Next Header		Yes

Proto 43 and 44 on IPv4?

- Protocols 43 and 44 have no defined meaning under IPv4
  - But under IPv6 they code for Fragment and Routing extension headers
- Is this usable?
- Or useful to an attacker?
- Inferring meaning from the lack of a Protocol Unreachable is not necessarily reliable
  - But points to possible areas of interest
- In certain Vista Beta 2 builds:
  - IPv4 packet with proto 43 caused BSOD
  - IPv4 packet with proto 44 caused partial unresponsiveness

Tunneling

- More tunneling is available in Vista than XP
- Now apparently have:
  - v[46] over v[46], Teredo, GRE, IPsec tunnel mode
- This is an area of concern due to possibility of security controls being bypassed
- On Vista, the Teredo component requires an IPv6 firewall be in place before it starts up
  - Appears to be same same safety check for IPv4 over IPv6
- We've been studying Teredo (more later)

ICMP Error Rate Limiting

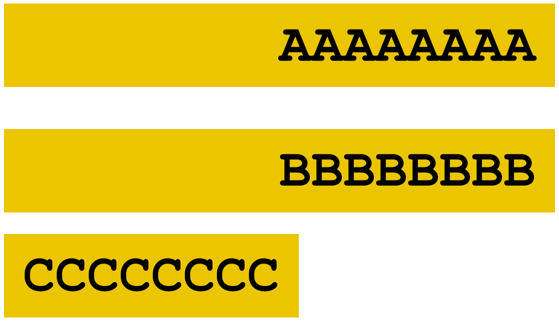
- Vista rate limits ICMPv4 and ICMPv6 error messages
  - Something like no more than one per second
  - RFC 2460 requires some kind of rate limiting for ICMPv6 errors
- So, had to slow down IP proto and UDP port scanning
  - Since those depend on ICMP error messages
- It was useful to use virtualization (e.g., VMware) to have extra copies of target to save time

IP Fragment Reassembly

- Empirically studied how Vista does IP fragment reassembly
- Found that Vista’s IP fragment reassembly is different from XP (or any other stack)
  - However, Vista's IPv4 and IPv6 have same behavior
- Means NIDSs will have to implement new strategy to prevent evasion attacks

IP Fragment Reassembly

- Two fully overlapping segments



AAAAAAAA

BBBBBBBB

CCCCCCCC

- Windows Vista and XP: prefer previous data (favor old)

CCCCCCCCAAAAAAAA

- Linux: favor new

CCCCCCCCBBBBBBBB

IP Fragment Reassembly

- Two partially overlapping fragments

BBBBBBBBBBBBBBBBB

AAAAAAAAAAAAAAAAA

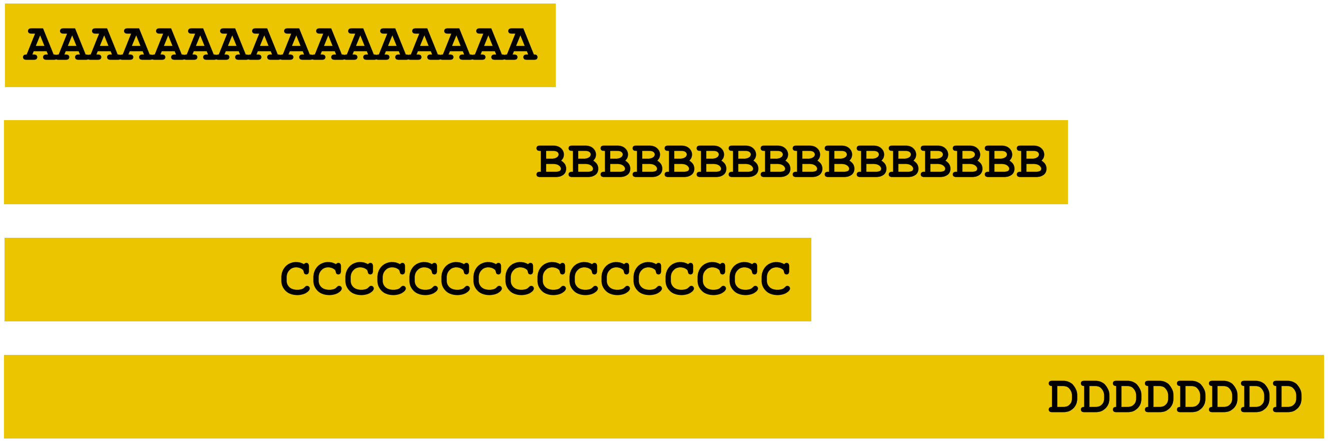
- XP: reassembled as:

AAAAAAAABBBBBBBBBBBBBBBBB

- Vista: no reassembly

IP Fragment Reassembly

- Vista fragment reassembly can succeed with partial overlap
  - However, the overlap must occur within the part of the packet that could already be assembled, starting from offset 0
  - The new fragment is ignored



AAAAAAAAAAAAAAAA

BBBBBBBBBBBBBBBB

CCCCCCCCCCCCCCCC

DDDDDDDD

- Reassembled:

AAAAAAAAAAAAAAAABBBBBBBBBBBBBBBBDDDDDDDD

- Doesn't seem like reassembly behavior is based on intentional policy decision
- More details in paper

Observing IPv4 Fragment Reassembly

Somehow, we need to observe how the packet is assembled

IPv4 test cases:

- The region that is fragmented ambiguously is a UDP payload
- Set up a UDP socket (nc -u -l) on the recipient system that the recipient stack passes the reassembled packet to
- UDP checksum set to 0 (no checksum) to avoid presumption of how the UDP packet will be reassembled

This doesn't work for IPv6 since UDP checksum is required

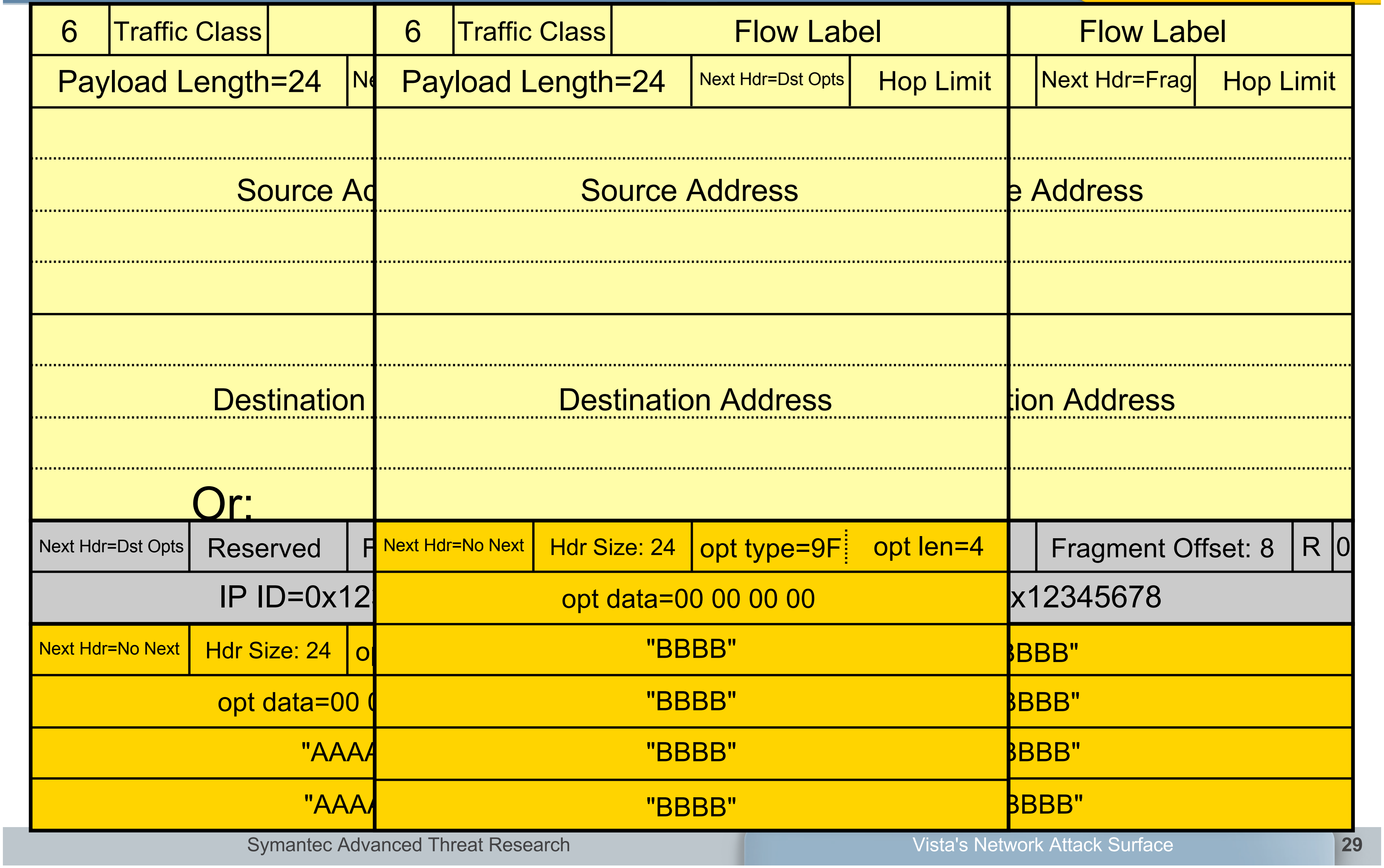
- So, we had to develop a new approach

Observing IPv6 Fragment Reassembly

IPv6 test cases:

- When returning an ICMPv6 error message in response to a packet, RFC 2460 (IPv6 spec) requires the full “original” packet be included (up to 1280 octets in return packet)
  - We take advantage of this
- We use the approach of sending a packet that when reassembled will (by design) yield an ICMPv6 error
  - So we can see how it was reassembled
- We used a destination option with option type 0x9F
  - No such type has been defined but type is 10xxxxxx so RFC 2640 requires an ICMP error message if it is not understood

Assembling the IPv6 Fragments



6 | Traffic Class | 6 | Traffic Class | Flow Label | Flow Label

Payload Length=24 | Ne | Payload Length=24 | Next Hdr=Dst Opts | Hop Limit | Next Hdr=Frag | Hop Limit

Source Ac | Source Address | e Address

Destination | Destination Address | tion Address

Or:

Next Hdr=Dst Opts | Reserved | F | Next Hdr=No Next | Hdr Size: 24 | opt type=9F | opt len=4 | Fragment Offset: 8 | R | 0

IP ID=0x12 | opt data=00 00 00 00 | x12345678

Next Hdr=No Next | Hdr Size: 24 | o | "BBBB" | 3BBB"

opt data=00 (| "BBBB" | 3BBB"

"AAAA | "BBBB" | 3BBB"

"AAA/ | "BBBB" | 3BBB"

Agenda

1 Introduction

2 Windows Vista Firewall

3 Layer 3

4 Layer 4

5 Teredo

6 Additional results

TCP Port Enumeration

Scanning from same subnet when set to private profile:

TCP Port/Protocol	IPv4	IPv6
Almost all ports	Filtered (no response)	Filtered (no response)
5357/Web Services on Devices	Open (SYN-ACK)	Open (SYN-ACK)

TCP Port Enumeration (Firewall Off)

TCP Port/Protocol	IPv4	IPv6
135/RPC endpoint mapper	Open (SYN-ACK)	Open (SYN-ACK)
139/NBT	Open (SYN-ACK)	Closed (RST)
445/SMB	Open (SYN-ACK)	Open (SYN-ACK)
5357/Web Services on Devices	Open (SYN-ACK)	Open (SYN-ACK)
49152-49157/RPC ephemeral	Open (SYN-ACK)	Open (SYN-ACK)

(default ephemeral port range is different than XP)

UDP Port Enumeration

Scanning from same subnet when set to private profile:

UDP Port/Protocol	IPv4	IPv6
All ports	Filtered or open (no response)	Filtered or open (no response)

Based on firewall rules state and netstat, these may be open for IPv4 and IPv6:

- 137/NetBIOS name service (IPv4 only)
- 138/NetBIOS datagram
- 3702/Web Services Discovery
- 5355/LLMNR

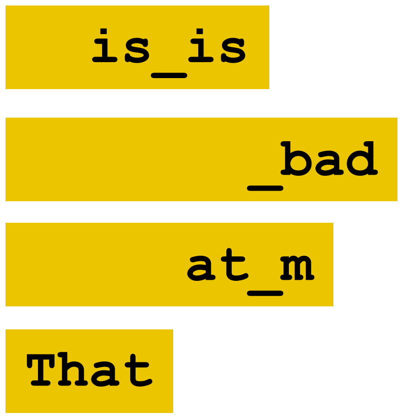
UDP Port Enumeration (Firewall Off)

UDP Port/Protocol	IPv4	IPv6
123/NTP	Open	Open
137/NetBIOS name service	Open	Closed (ICMPv6 Port Unreachable)
138/NetBIOS datagram	Open	Closed (ICMPv6 Port Unreachable)
500/ISAKMP	Open	Open
1900/UPnP/SSDP	Open	Open
3702/Web Services Discovery	Open	Open
4500/IPsec	Open	Closed (ICMPv6 Port Unreachable)
5355/LLMNR	Open	Open
3-4 variable ephemeral ports	Open	Open

(Some open ports are clients)

TCP Segment Reassembly

- Four overlapping segments:



```
  is_is
       _bad
     at_m
That
```

- Vista: This_is_bad
- XP: That_is_bad
- Linux: That_is_mad

- Old data is always preferred over newer data
- Behavior is novel
  - NIDSs will have to implement new strategy to prevent evasion attacks

Agenda

1 Introduction

2 Windows Vista Firewall

3 Layer 3

4 Layer 4

5 Teredo

6 Additional results

Microsoft IPv6 Transition Mechanisms

To allow more clients to use IPv6 on the Internet, Microsoft has implemented transition mechanisms for IPv6, including

- ISATAP
  - IPv6 directly on top of IPv4
- Teredo
  - IPv6 on top of UDP over IPv4
  - Developed by Christian Huitema of Microsoft
  - Published as RFC 4380: Tunneling IPv6 over UDP through NATs

Teredo Introduction

- Teredo was developed because ISATAP/6to4 doesn’t work through IPv4 NATs
- It is supposed to be an IPv6 provider of last resort
  - Only when native connection or ISATAP not available
  - Definitely more overhead than native IPv4 or IPv6
- Provides host to host automatic tunneling
- Provides automatic IPv6 address assignment
- Doesn’t require support of local network at all

Teredo on Windows Vista

Teredo is enabled by default in Windows Vista

- It is the IPv6 provider of last resort
- However, it may not always be the IP provider of last resort
  - Teredo may be used in favor of native IPv4 in some circumstances
- When Teredo is used is a complicated topic
  - Depends on application behavior
  - MS documentation is unclear
- Safest to assume that the Teredo interface will often be active

CVE-2007-1535/BID 23267: Inaccurate Teredo Use Documentation

- Microsoft’s “Teredo Overview” and other pages say (emphasis mine):
  - “In Windows Vista, the Teredo component is enabled but inactive by default. In order to become active, a user must either install an application that needs to use Teredo, or configure advanced Windows Firewall filter settings to allow edge traversal.”
  - (http://www.microsoft.com/technet/prodtechnol/winxppro/maintain/teredo.mspx)
- Not in our experience with Vista RTM:
  - We have some Vista hosts that are normally connected to isolated network (not for Teredo research)
    - No added applications or firewall changes
  - When one was accidentally attached to an Internet network during Vista installation, it had set up a Teredo address before we knew it was Internet-connected
  - Separately, when connected to an Internet network to complete Windows Activation, Vista obtained a Teredo address
  - Isolated incidents?
  - Also, “ping -6” will cause Teredo to be used (if no other IPv6 access)
- Documentation remains unfixed

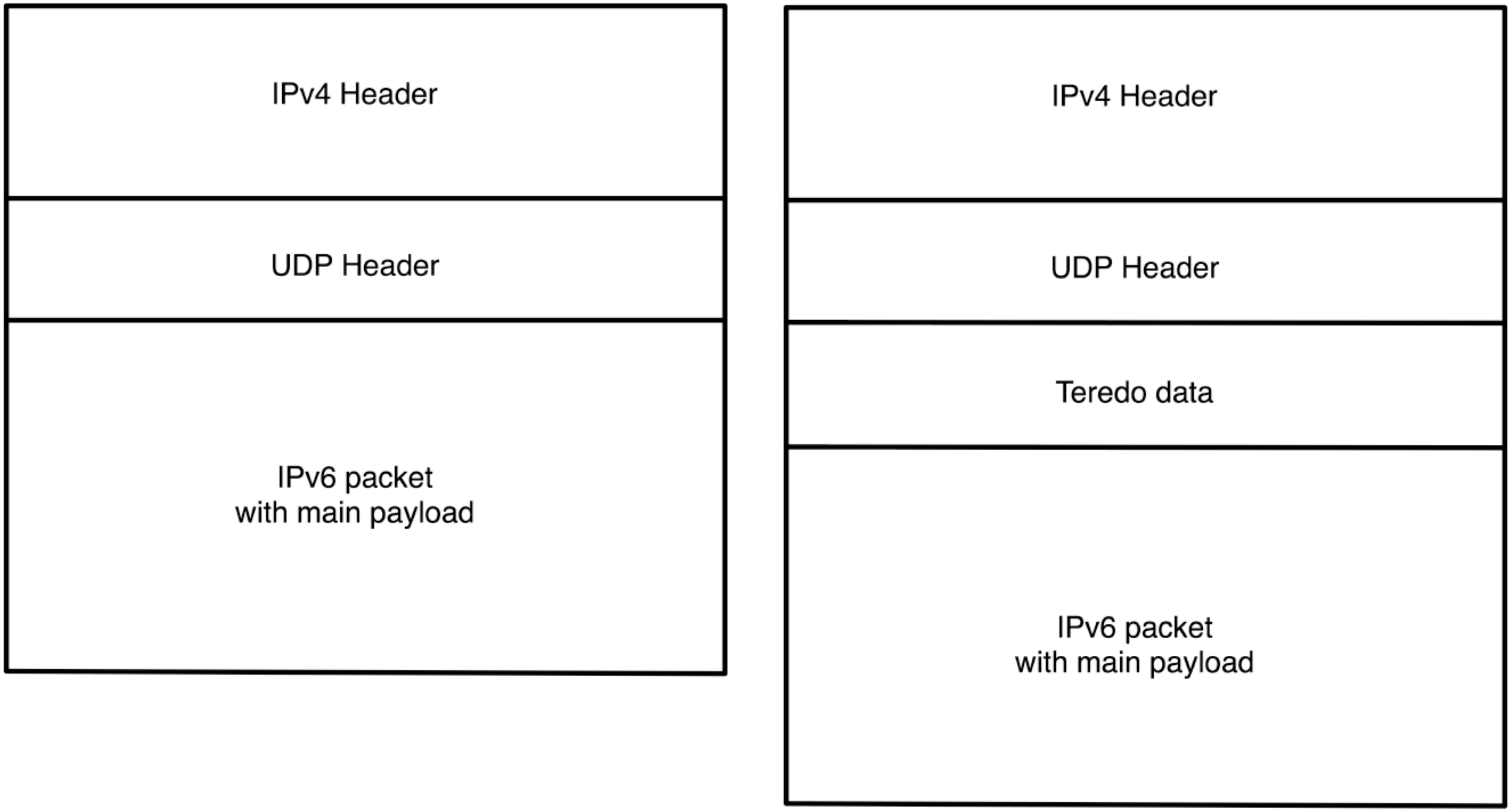
Teredo Security Research

- Teredo’s ability to traverse NATs caught our eye in terms of security so we did some research
- First did some platform-independent analysis of security implications
  - Studied RFC 4380
  - Found some implications that weren’t documented
  - Wrote paper: *The Teredo Protocol: Tunneling Past Network Security and Other Security Implications*
- Then we looked at the Vista Teredo implementation

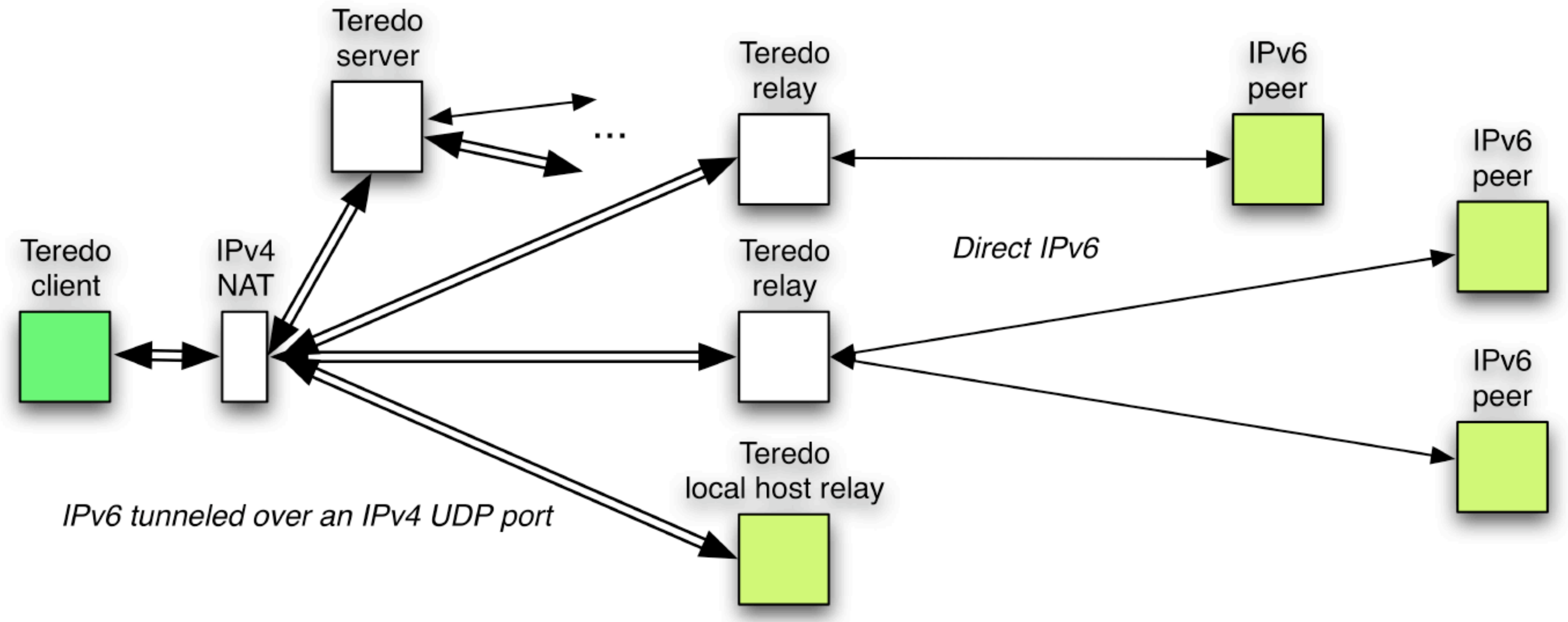
It would take > 1 hour to explain how all the Teredo protocol works

- See Teredo paper for an introduction

Teredo Encapsulation



Teredo Tunneling



- IPv6 packets are encapsulated in UDP and IPv4 for the part of the route that is over IPv4
- Rest of route is native IPv6
- Peer host need not be aware of Teredo
  - Relays (on the Internet or on the peer) encapsulate and decapsulate packets between Teredo client and IPv6 peer

Teredo Server and Addresses

Teredo server helps a client set up a Teredo address and find a relay for an IPv6 peer

- The server to use is usually statically configured on client

Teredo addresses are global scope

- Packets can be sent from anywhere on Internet and reach the client
- All start with 2001:0000::/32

Teredo address format:

```
+---------------+---------------+-------+-------+---------------+
| Prefix        | Server IPv4   | Flags |C. Port| Client IPv4   |
+---------------+---------------+-------+-------+---------------+
| 32 bits       | 32 bits       |16 bits|16 bits| 32 bits       |
|                               |                               |
```

Note: last 2 fields have their each of their bits flipped

Teredo Main Security Concerns

- Teredo puts hosts directly on Internet (with a stable open-ended tunnel)
  - Global addressability is the way it is supposed to be with IPv6
  - However, with native IPv6, admins would be aware of it
  - With Teredo, hosts will be unexpectedly exposed
    - Even if have a private IPv4 address and are behind a NAT
- Teredo also bypasses inspection by network security devices (e.g., firewall, network IPS)
  - Unless they are specifically Teredo aware
  - Some important security controls provided by the network may not be in place on end-host
  - Defense in depth is reduced in any case
- Vista:
  - Teredo might often be active
  - Windows Firewall is applied to tunneled Teredo packets

Security Concern: Cost To Find All Teredo Packets

- Inspecting contents of all Teredo packets on the wire is not trivial
  - Only server-bound traffic has a characteristic port (UDP 3544)
  - Relay and clients can be on any port
  - So, need to apply a heuristic to all packets on all UDP ports
    - Can be expensive

- Blocking outbound port 3544 should *eventually* starve normal Teredo clients of ability to connect
  - Especially if blocking is done between client and it's NAT
  - May not prevent outbound malicious or intentionally evasive connections though

Security Concern: Teredo Information Disclosure

- Non-concerns (probably):
  - Teredo servers don’t process real traffic (only set-up packets)
  - Teredo relays– not any different than typical router
- However, server knows (essentially) all of client’s peer IPv6 addresses
  - Okay if you trust the server not to make bad use of it
  - Vista and XP use Microsoft operated servers by default
    - Any conspiracy theorists out there?
    - Can probably trust Microsoft on this
- Also, a Teredo address has some info that can be used to profile address owner ahead of time

Security Concern: Teredo Server Bumping (1)

What if some malware or malicous user changed a host's setting for what Teredo server to use?

- Assuming the new server functions mostly properly, user is unlikely to notice
- However, the new server could be malicious
- Could snoop what your peer hosts are
- If you ask a malicious Teredo server to help you find a relay for an IPv6 server, it can lie and say that *it* is the correct relay to use
  - It can also have a separate host respond to you as the fake relay
  - Various uses in phishing/pharming similar to changing DNS server setting

Security Concern: Teredo Server Bumping (2)

How much of a concern?

- Depends on if the implementation prefers Teredo to native IPv4
- Potential for the server to spoof all IPv6 capable servers on Internet (and any other IPv6 peers)

Vista:

- Need admin privileges to change Teredo server setting
- If you try to read Teredo server setting as a non-admin, it'll say "teredo.ipv6.microsoft.com" regardless of the actual setting
  - So easier to miss the changed server
  - Also always says that Teredo is not being used

Teredo Security Positives

- RFC 4380 requires a lot of sanity checking on packets
  - Prevents a number of attacks
- Decent anti-spoofing mechanisms used
  - Beneficial for the case where IPsec is not being used
  - However Vista uses a substandard “ping nonce” strength (32 instead of recommended 64 bits)
    - Slightly increases chance of peer spoofing
- Can use IPsec in normal manner
  - Hard to use IPsec with 6to4

Teredo Suggestions

There are additional Teredo security concerns: see the Teredo paper

I recommend:

- Disable Teredo and block on network
- Upgrade security controls and posture to support native IPv6
- Only then, obtain a native IPv6 connection to the Internet

Agenda

1 Introduction

2 Windows Vista Firewall

3 Layer 3

4 Layer 4

5 Teredo

6 Additional results

Default Source Routing Behavior on Vista

- Based on netsh and tests

Kind of source routing encounter	Native IPv4 (LSRR)	Native IPv6 and Teredo (routing type 0)
En route (more hops follow)	Will not forward	Will not forward
At end (we are last hop)	Packet discarded	Packet accepted

Vista’s Default IPv4 ID Range

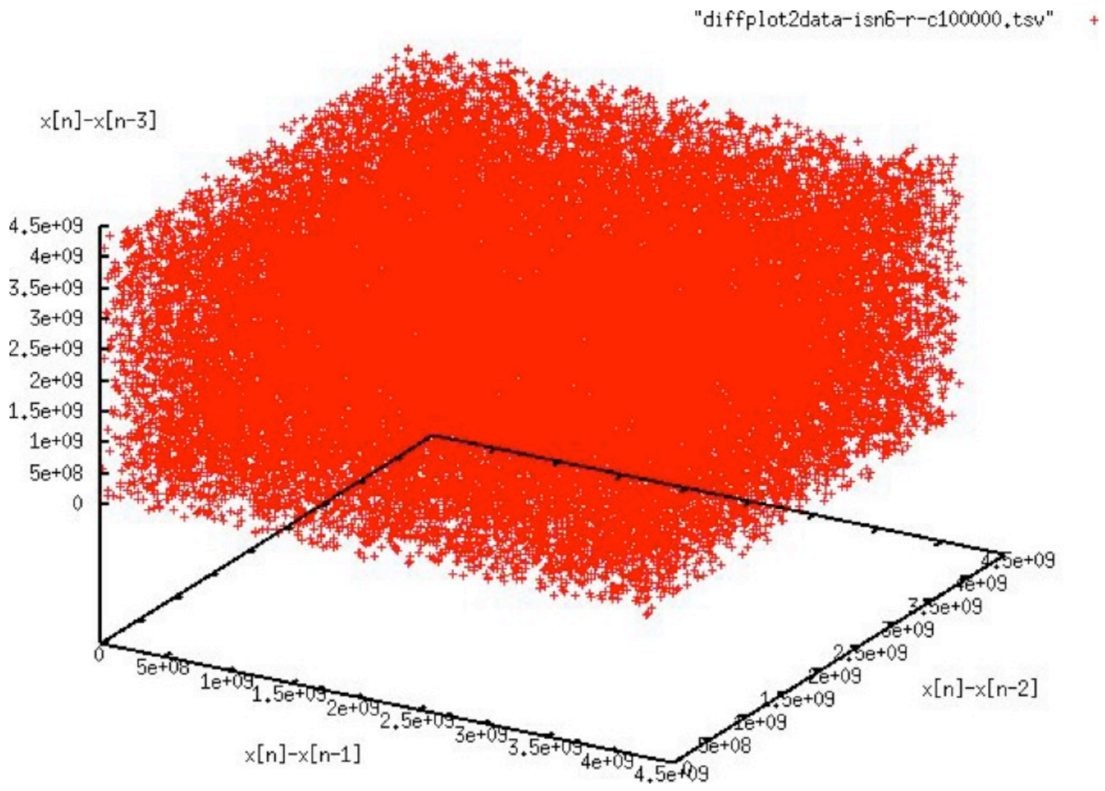
- IPv4 ID range used is 0 to 0x7fff and used incrementally
  - Uses half the available range
  - Should still be able to do host counting behind a NAT

TCP ISN Generation

- Choice of TCP initial sequence number affects attacker’s ability to blindly attack a connection
- Vista (like XP) looks like it follows RFC 1948 for both IPv4 and IPv6
- Nmap:

```
TCP Sequence Prediction: Difficulty=261 (Good luck!)
```

Plotting TCP ISN Generation



- 100,000 TCP packets over IPv6
  - Record ISN as x[i]
- Plot <x[i]- x[i-1], x[i]- x[i-2], x[i]- x[i-3]>
- Looks uniform

- Ditto for IPv4 and plotting <x[i]- x[i-1], x[i-1]- x[i-2], x[i-2]- x[i-3]>

ARP and ND Attacks

- Attacker can cause false IPv4/6-MAC assoc. in some cases
  - A.k.a. cache poisoning (enables man-in-the-middle, DOS)

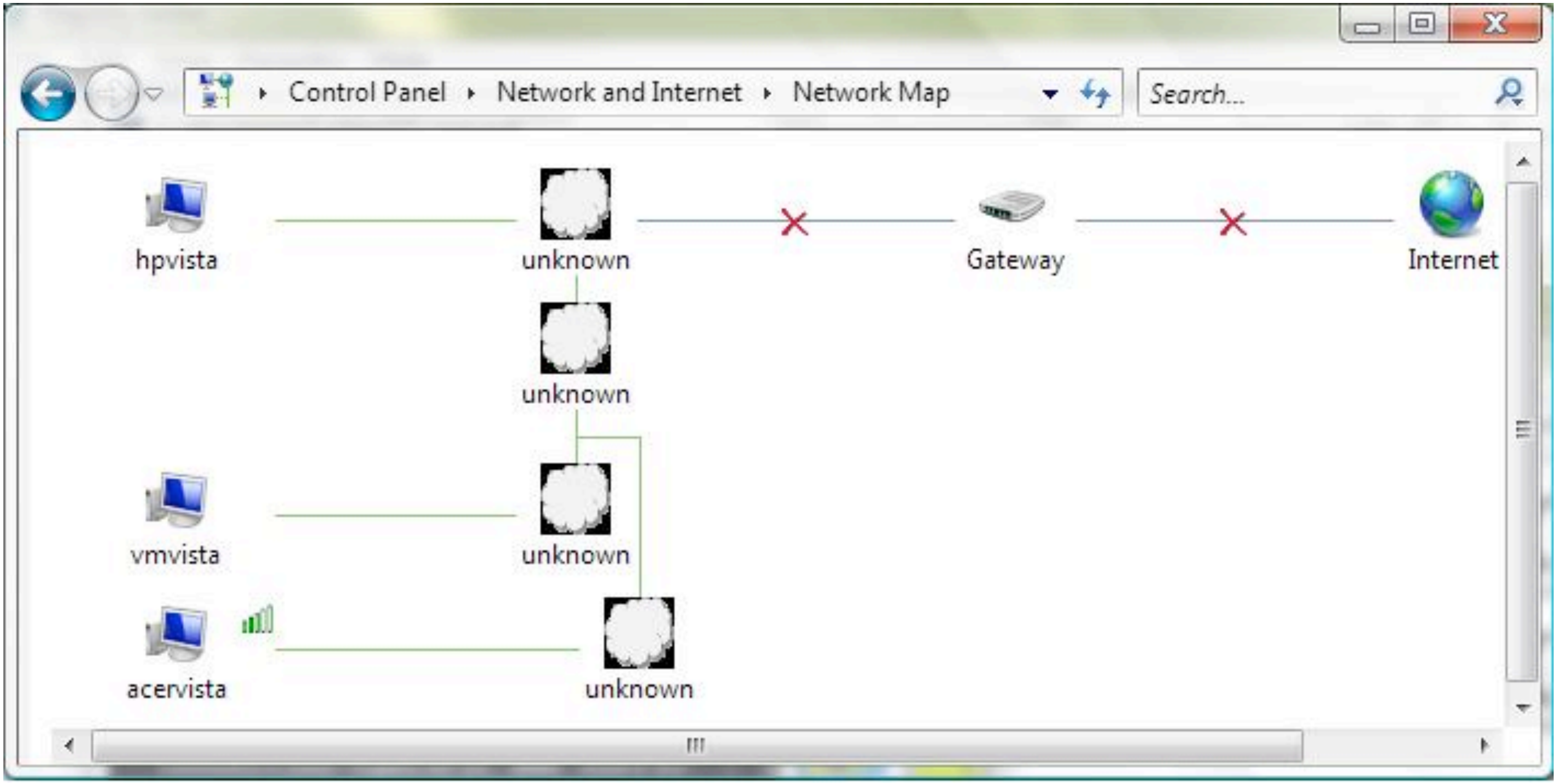
Attack	ARP (IPv4)	ND (IPv6)
Fake an upd. to an existing entry	Will overwrite and be used	Will overwrite and be used
Unsolicited fake assoc. for address with no entry	Not stored or used	Not stored or used
Solicited false reply for address with no entry (directed reply)	Creates entry and gets used	Creates entry and gets used
Solicited false reply for address with no entry (broadcast/multicast reply)	Not stored but will be used if needed	Creates entry and gets used
Faked address conflict	Statically configured address: interface becomes unusable until reset, like XP	Link-local RFC 3041 address: automatically generates new address

LLTD Research

- We looked into LLTD protocol and Vista's implementation of it
- Performed on July CTP build 5472 (not updated for RTM)
- Purpose of the research:
  - Understand the LLTD protocol
  - Any security implications which would arise from its deployment
  - Identify any implementation issues within Microsoft's implementation

Link Layer Topology Discovery

- Network mapping for diagnostics



LLTD Research Conclusions

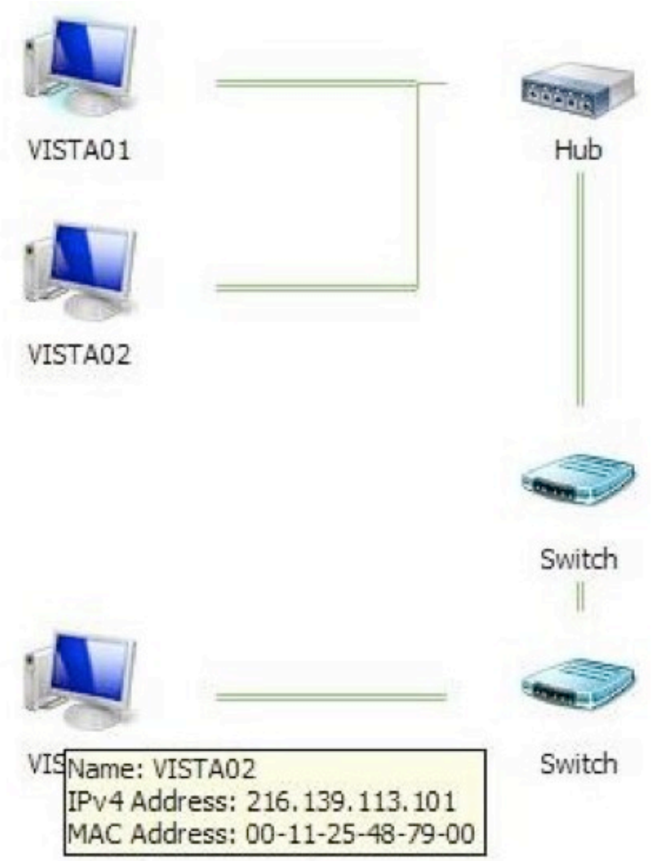
Conclusions

- LLTD is a simple non routable protocol
- Even if a vulnerability were discovered it would require an attacker to have local LAN access to exploit
- Little exposure for corporate or home networks
- Evidence of Microsoft’s SDL (Security Development Lifecycle) throughout the protocol design and implementation

LLTD doesn't raise many concerns, however:

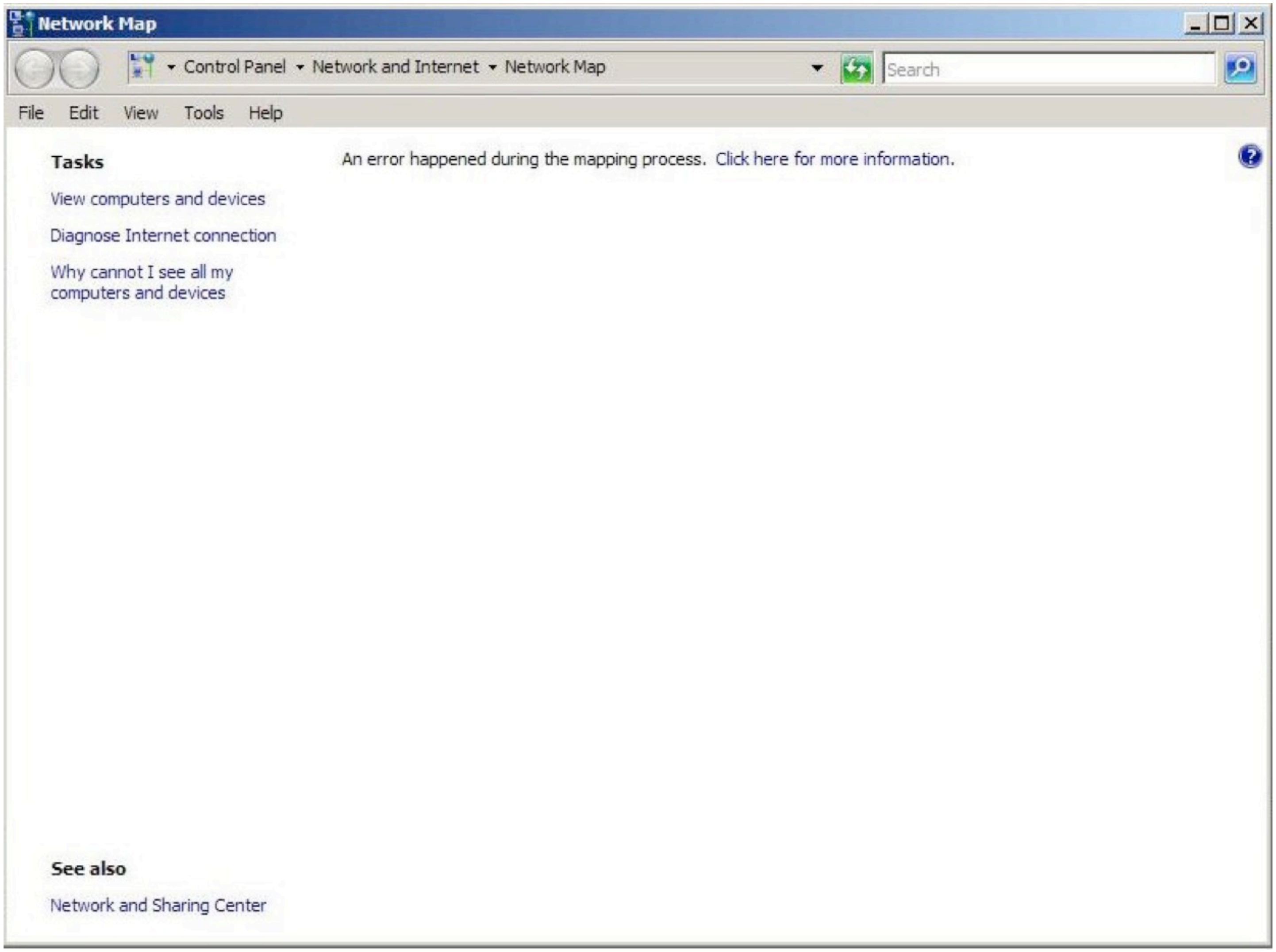
- It could be used in recon
- It is pretty easy to add fake data to map from local network
  - Including that an address has a web-based management interface
    - Can use to unexpectedly direct someone to Internet host from right-click
  - Can provide icon to display
- Also easy to DoS network mapping

Example of Faking Data on Network Map Using LLTD

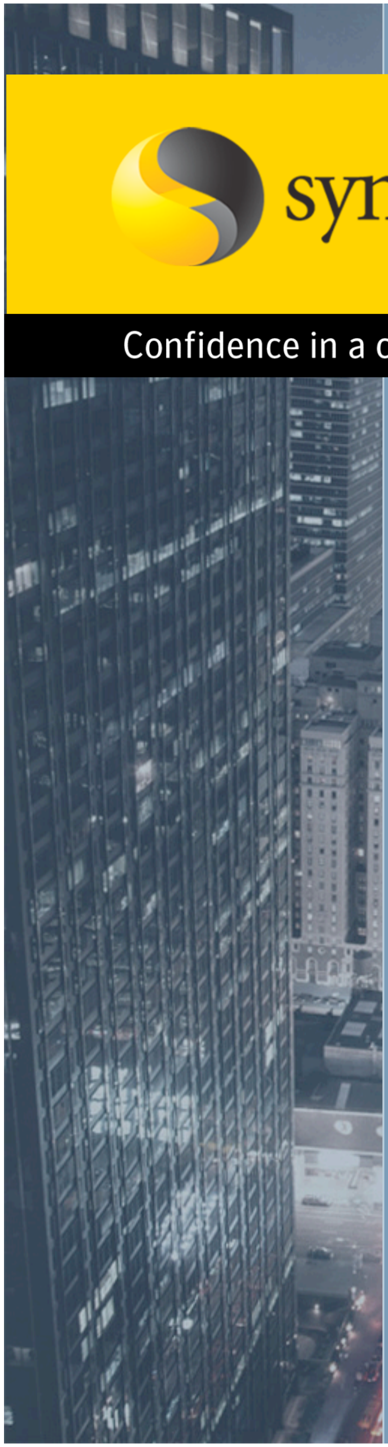


Name: VISTA02
IPv4 Address: 216.139.113.101
MAC Address: 00-11-25-48-79-00

DoS of Network Mapping with Malicious LLTD Responder


Network Map
Control Panel ▸ Network and Internet ▸ Network Map
Search
File Edit View Tools Help
Tasks
View computers and devices
Diagnose Internet connection
Why cannot I see all my computers and devices
See also
Network and Sharing Center

An error happened during the mapping process. Click here for more information.



symantec™

Confidence in a connected world.

Questions?

symantec™

Confidence in a connected world.

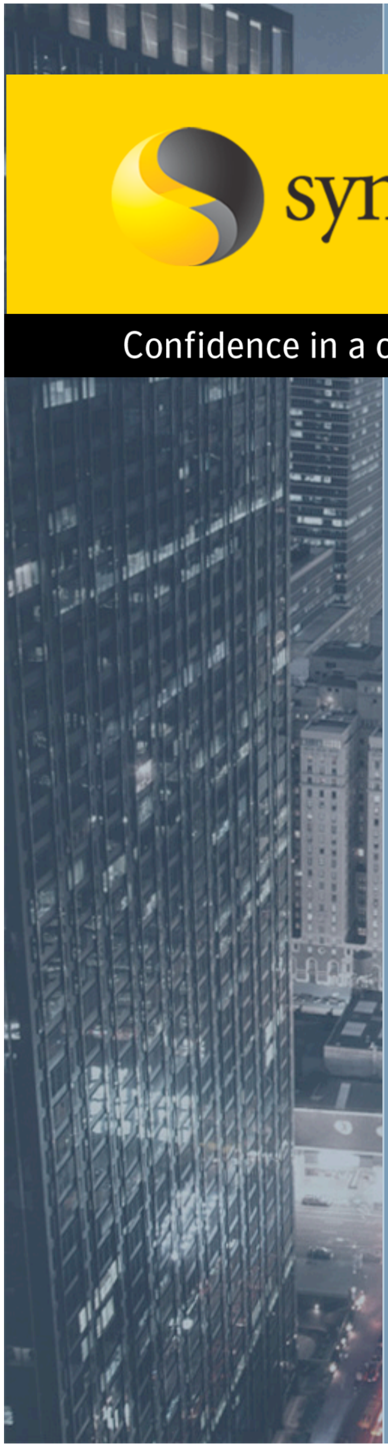
Thank you!

Jim Hoagland

jim_hoagland@symantec.com

http://www.symantec.com

This document is provided for informational purposes only and is not intended as advertising. All warranties relating to the information in this document, either express or implied, are disclaimed to the maximum extent allowed by law. The information in this document is subject to change without notice.

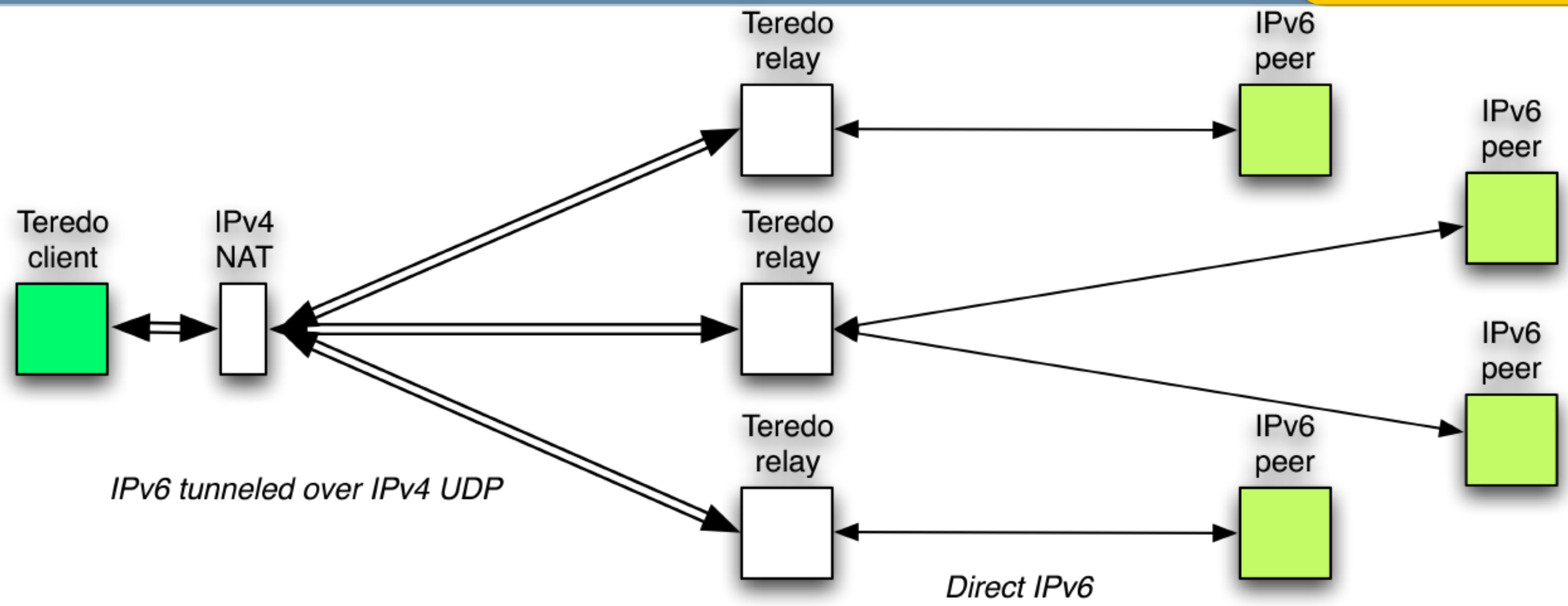


symantec™

Confidence in a connected world.

Backup Slides

Teredo Relay

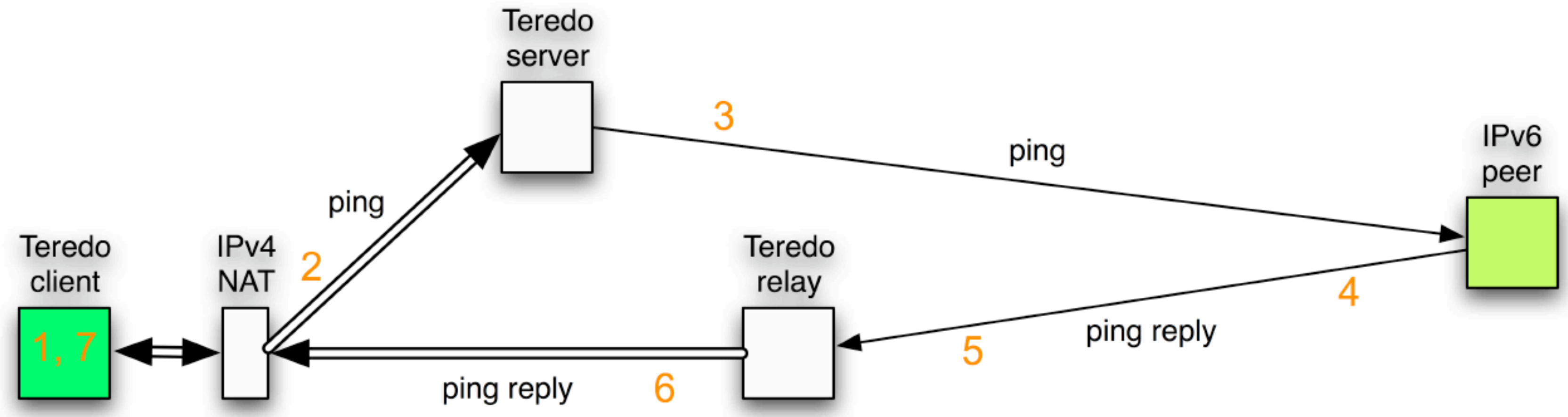


Using a relay, both Teredo clients and peers can initiate a packet send

- Native IPv6 peer finds relay since relay advertises a route to 2001:0000::/32
  - Teredo addresses contain enough information for relay to reach Teredo client by IPv4

- Teredo client finds relay to use with help from Teredo server
  - Ping test establishes what relay will be used to reach a peer
  - Also used to guard against peer spoofing

Ping Test Procedure

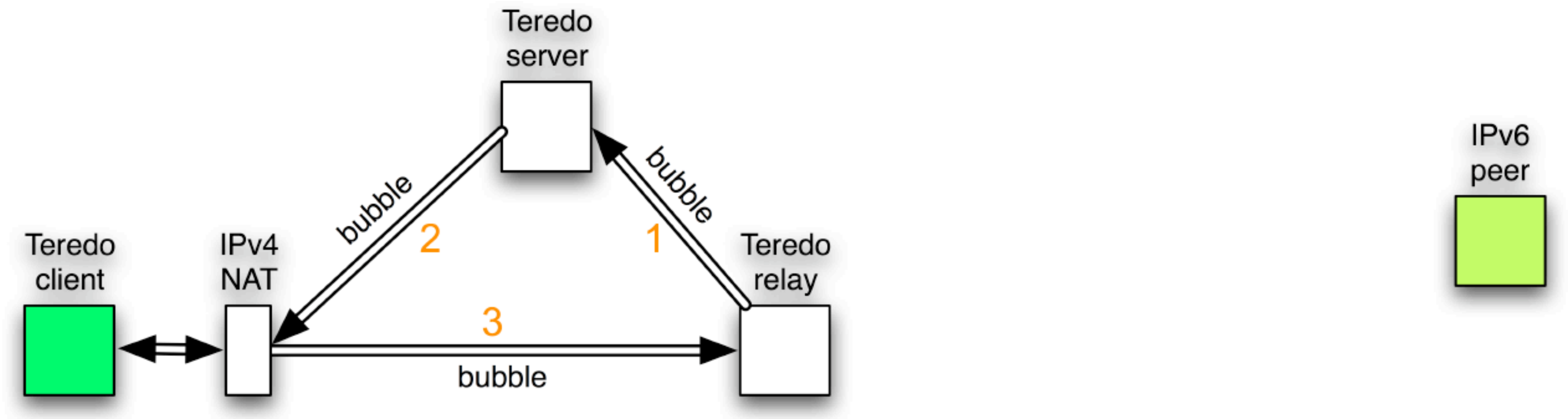


Ping test procedure (for any new peer):

1. Client creates an IPv6 echo request (ping) addressed to the peer
   - Payload is a random number (nonce)
2. Client encapsulates this and sends to its server
3. Server decapsulates and drops on the IPv6 Internet
4. Peer responds to ping
5. Echo reply is routed to nearest relay
6. Relay encapsulates this and provides passes to client
7. Client inspects echo reply
   - Verifies nonce payload matches what it sent (reply was not spoofed)
   - Client remembers source IPv4 address and port as relay to use for peer
   - Also as the only address to accept packets from for peer

Relay Bubble Procedure

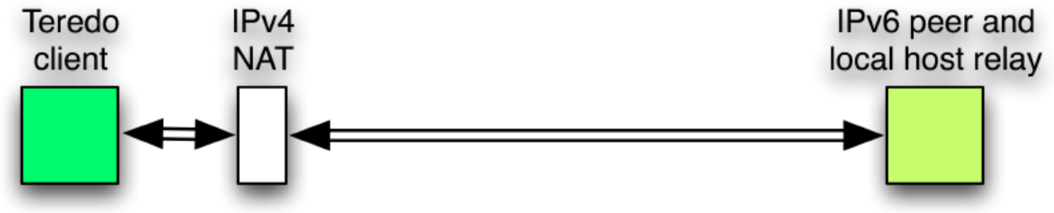
- Some NATs won't allow packets to come in on client's Teredo port unless it is a recent outbound destination
- Relay needs to work around this before it can pass along the echo reply



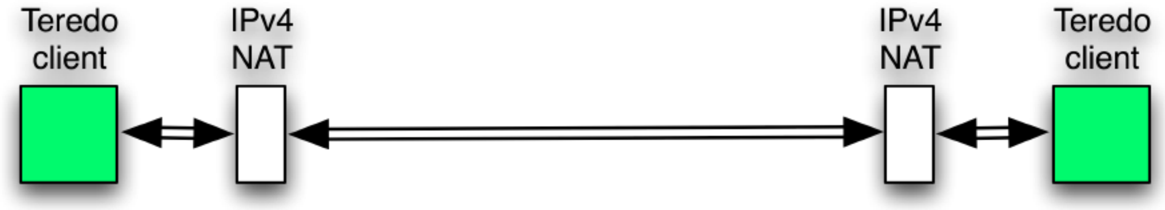
- Relay sends a "bubble" (empty IPv6 packet) to the client's server, asking the server to pass it along to the client and to ask the client to send it back to relay
  - Thus the relay becomes a recent outbound destination (defeating the NAT's restriction)
  - Server is a recent destination due to regular packets sent by client

May Not Need an Internet-based Teredo Relay

- If IPv6 peer has global IPv6 and IPv4 addresses and is Teredo-aware, it can be its own “local host relay”
  - Packet is encapsulated before leaving peer
  - Tunneled for full route (no IPv6 networks needed)
  - Vista and Longhorn: serve as local host relays when they have a native IPv6 address



- Teredo client to Teredo client communication also takes this shortcut



Security Concern: Teredo Address Scanning

- Teredo addresses are much easier to guess than native IPv6
  - Fields can be pretty predictable
- Thus blind address scanning may be feasible
  - Unlike general IPv6 case
- Some public IPv4 addrs will have many ports open for Teredo clients
  - E.g. external NAT IPs for large organizations and for ISPs that only provide private IP addresses
  - Makes it easier to guess a Teredo client for the IPv4 address
  - Also makes Teredo addresses for that locality easier to guess
- Vista adds in 12 random bits in address (flags field)
  - This makes addresses 4096 times harder to guess
  - Note: actual randomness of the 12 bits hasn’t been studied
- Vista clients:
  - Server field is pretty predictable
  - Client port number drawn from 49152-65536
    - Will sometimes make external port number more predictable

Security Concern: Teredo + Source Routing

- What if Teredo-tunneled IPv6 packet specifies source routing?
  - Teredo client might well forward the IPv6 packet after decapsulating it
  - Could forward an IPv6 packet to an internal host (or to an external host)
  - That would bypass router source-routing controls
  - Vista: doesn't forward source routed packets by default

More Teredo Security Concerns

- Worm impact
  - The main benefit to a worm from Teredo is ability to reach through NAT to end host
  - A worm that exploits Teredo implementation or anything pre-security could be really bad
- If peer-to-peer (e.g. PNRP) enabled, inbound packets would be allowed
- There are a number of possible ways to take out Teredo service for a host or for part of the Internet
- Teredo's bubble-to-open mechanism effectively converts a restricted NAT into an unrestricted one, for the Teredo port
- And more…

More Observations from Vista

- Vista prefers to use new version of SMB, SMB2
- Successfully calling RPC procedures over SMB named pipes varied between XP (SMB) and Vista (SMB2) callers
  - Even within an interface

Crash 1 from ISIC

- IPv4 packet with IP protocol # 43 and random payload
- Beta 2 build 5270: Blue screen
- Proto # 43 undefined in IPv4 but in IPv6 it is the Routing extension header
  - Aside from a handful of extension headers, IPv6 next header values are the same as IPv4 protocol values
  - So, stack may have used shared lookup table
- Results in attempt to read memory at 0x00000002

Crash 2 from ISIC

- IPv4 packet with protocol # 44 and random payload
- Beta 2 build 5270: Target becomes partially unresponsive
- Proto # 44 undefined in IPv4 but in IPv6 it is the Fragment extension header
- Exact reason for hang not clear

Crash 3 from ISIC

- IPv4 option field: 95 00 00 00
    - Option field is a list of options in TLV format
    - Option type=0x95 (undef)
    - Length = 0 (illegal, should be ≥2)
- Beta 2 build 5270: Target become locked up until reset
- Maybe infinite loop (stuck processing start of options over and over)

Historic Layer 3/4 DoS Attacks

- Had some successful attacks in beta builds (only tried IPv4)
- Blat
  - SYN flood with URG pointer pointing past end of packet
  - Network stack was unresponsive for a few seconds
- Land
  - SYN with source IP=destination IP
  - Attempt to cause host to reply to itself
  - Network stack was unresponsive for a few seconds
- OpenTear
  - Invalid UDP fragments
  - Sent from many source addresses
  - Network stack was unresponsive for the attack duration

MS-RPC Named Pipes Over File Sharing

- Windows allows RPC access via named pipes over SMB/SMB2/CIFS
  - Via IPC$ share
  - We wanted to enumerate the pipes available via null and authenticated sessions
- File sharing is disabled by default on Vista, so we enabled it
- Start by using pipelist.exe (sysinternals) on Vista to find all local named pipes
  - Also looked at HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\LanmanServer\Parameters\NullSessionPipes and HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Npfs\Aliases
  - And at endpoint mapper information
- Tried opening each pipe for read/write from both Vista (SMB2) and XP (SMB)

Remotely Accessible Named Pipes

Via null session from XP or Vista:

- netlogon
- lsarpc
- samr

Via authenticated session from XP or Vista:

- atsvc
- browser
- DAV RPC SERVICE
- epmapper
- eventlog
- InitShutdown
- keysvc
- lsarpc
- lsass
- LSM_API_service
- MsFteWds
- Netlogon
- ntsvcs
- plugplay
- protected_storage
- ROUTER
- samr
- scerpc
- srvsvc
- tapsrv
- trkwks
- W32TIME_ALT
- wkssvc

Calling RPC Via Named Pipes

- Then we wanted to see what RPC interfaces and procedures could be accessed over the named pipes
- Developed a list of interface UUIDs so can try all
  - Had a list from previous testing
  - Add all UUIDs seen from endpoint mapper
  - Also did static binary analysis on Vista executables to find additional UUIDs
    - Include UUIDs seen from both client and server side
- Procedures on an interface could be called by number
  - Can get name from available symbols
  - Calling procedures blindly so BAD_STUB_DATA is same as success

- We found that RPC access is selective
  - Not all pipes have same access to interfaces
  - Not all procedures in an accessible interface are accessible in all circumstances

RPC Procedures Callable Via Named Pipes

- Found we could call 102 procedures on five interfaces via null session
  - All three named pipes have same access
  - XP and Vista had same access
- Could call 338 procedures on 15 interfaces from authenticated XP
- But only 229 procedures on 10 interfaces from authenticated Vista
  - Mostly due to six interfaces that were XP-only
  - Some interfaces had both XP-only and Vista-only procedures
- May mean that that there is different code handling
  - Or perhaps that access is selective on XP/SMB vs. Vista/SMB2
- 6bffd098-a112-3610-9833-46c3f87e345a (in wkssvc.dll):
  - XP could call procs 0-31 from multiple named pipes
  - Vista could only call procs 0-30
  - Can’t find a name for proc 31 or see the code to handle it
- There are may be other procedures and UUIDs we don't know about

Background: IPv6 Options

- IPv6 has a fixed length base header
  - IPv4’s options have been moved to hop-by-hop and destination options extension headers (EHs) and other specialized extension headers
- As in IPv4, the options data is a packed sequence of TLV options
- Must be padded to make EH length a multiple of 8 octets
- Unlike in IPv4, the option length (“L” of TLV) codes for just the payload and not whole option
  - Previous Windows had a vulnerability with IPv4 options with too short of coded length

Testing IPv6 Options

- We hypothesized that there could be flaws in processing IPv6 options
- So, we sent random IPv6 packets with malformed destination options to Vista hosts
  - Random EH length (even larger than MTU), random EH payload
  - Sometimes starting the options with well-coded options
  - Sometimes starting with well-coded options whose types are 00xxxxxx (skip if not understood)
  - With nothing past the EH and with the packet being an ICMPv6 ping
- Tested certain combinations for 210 million packets with RTM
  - Didn’t observe any persistent problems

Testing IPv6 Options Sequentially

- Also tried a more orderly (precise) approach
  - Send a single option in a dest opts EH, plus any needed preceding pad octets
  - Three nested loops:
    - Option type (0..255)
    - Encoded option length (0..255)
    - Actual option length (before end of EH) (0.. encoded option length)
  - Random option payload
- For RTM:
  - Divided testing among four Vista hosts using ISIC-style -s (seed) and -k (skip sending) options to our test script
  - Tried all 8,421,376 combinations
    - No persistent effects noticed
  - Ran at one probe per second per host and in a ping packet (to avoid ICMP error rate limiting)
  - Hope to mine a data capture to find supported options and length
    - Using whether a probes produced an echo reply, or exact error returned
  - Extra hosts here sped up going through sequence space
    - Still took many days to complete

Ephemeral Ports

- Ephemeral port range is different than XP
- Vista uses 49152 to 65535, usually sequentially
- TCP often seen to use same port for IPv4 and IPv6
- UDP often seen to use adjacent ports for IPv4 and IPv6
- Range can be adjusted for TCP or UDP with netsh

SEcure Neighbor Discovery (SEND)

- Extension to NDP to eliminate many NDP security weakness
- CGA and public key for authentication (binding to address)
  - Difficult to use IPsec due to bootstrap
- Certification paths prove legitimacy of routers
- RSA signatures for integrity
- Timestamp and nonce for anti-replay
- Details in RFC 3971 and 3972

IPsec overview

- IPsec is a mandatory part of IPv6 so could be widely used
- Authentication Header (AH) is based on cryptographic checksum
- AH provides:
  - Integrity check (stronger than checksum)
  - Data origin authentication
  - Anti-replay services
- Encapsulating Security Payload (ESP) uses encryption
- ESP provides:
  - (all of AH functionality)
  - Message confidentiality
  - Traffic flow confidentiality (limited)
- Both are IPv6 extension headers

IPv6 Base Header

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version| Traffic Class |           Flow Label                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Payload Length        |  Next Header  |   Hop Limit   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                         Source Address                        +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                      Destination Address                      +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Extension Headers

- IPv6 base header now fixed at 40 octet size
- But there are now IPv6 extension headers

```
+-----------+-----------------+--------------------
| IPv6 base |  IPv6 extension |    next layer
|  header   |     header      |     protocol
|           |   (0 or more)   | (e.g. TCP,ICMP)
+-----------+-----------------+--------------------
```

- Next Header field indicates what is next
  - Same as IPv4 proto field but includes codes for EHs
- General format:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Next Header  |  Hdr Ext Len  |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               +
.                                                               .
.                       Extension  Data                         .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```