Vista Network Attack Surface Analysis and Teredo Security Implications

Dr. James Hoagland, Principal Security Researcher

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

Symantec Security Response – Advanced Threat Research

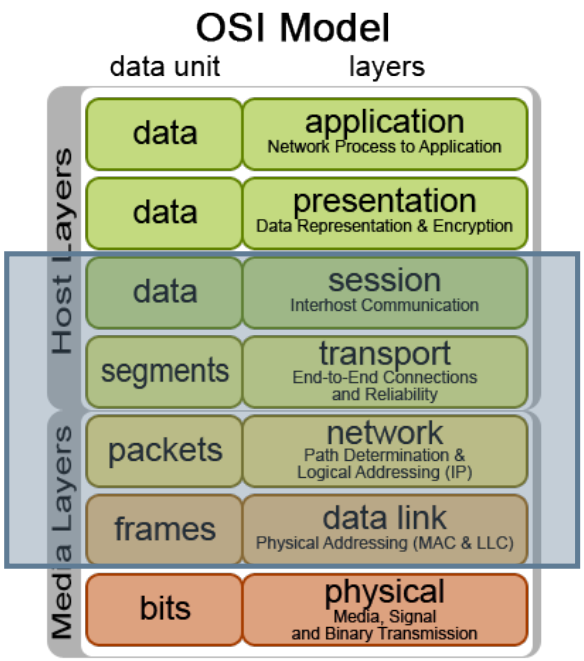
BlackHat Briefings, 2 August 2007

Main Take-Away from this Talk

- The network stack in Windows Vista is quite different than the one in Windows XP
  - So you may need to adapt how you do things as a result
- Teredo has a number of security concerns
  - Watch out for it tunneling on your networks

Windows Vista Network Attack Surface Analysis

- We examined the security-relevant aspects of Vista, from the point of view of the network
  - Part of Advanced Threat Research's review of Vista
  - Our motive: lots of systems will be running Vista so it's important to know what to expect
  - A very broad review, from layer 2 to 5
  - We dug fairly deep into some areas
- Results here are mostly from the out-of-the-box configuration with release (RTM) build of Vista
- Full details of this analysis are available in:
  - *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



Teredo Security Implications

- We also conducted analysis of the security implications of Teredo
- Teredo is a genus of shipworms
- Shipworms are not worms at all, they are considered mollusks
- Significant concerns for wooden ships, pilings, etc
  - They bore holes in wood
  - So you need to watch out for it

Teredo Security Implications (Take 2)

- We conducted a platform independent analysis of the security implications of Teredo
  - Teredo is an IPv6 transition mechanism that tunnels through NATs
  - It is enabled by default in Vista
- Full details of this analysis available in:
  - *The Teredo Protocol: Tunneling Past Network Security and Other Security Implications*
  - By Jim Hoagland
  - http://www.symantec.com/avcenter/reference/Teredo_Security.pdf

Outline

1. Introduction
2. Vista's new network stack and firewall
3. Some layer 3 & 4 results
4. IP & TCP reassembly behavior
5. The Teredo protocol
6. Teredo security implications
7. LLTD
8. Conclusion

What's New With Vista Networking

Some differences in Vista's networking that we'll discuss:

- Stack is a rewrite
- IPv6 is enabled and preferred by default
- IPv6 transition mechanisms present
- More tunneling mechanisms
- New Windows Firewall
- IP fragment and TCP segment reassembly
- Other different stack behaviors
- Other new protocols and exposures

New protocols and behaviors:

- Have implications for security devices
- Should influence enterprise policies and security controls

Microsoft Loves 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

The TCP/IP stack was rewritten in Windows Vista

- Partly to fully support IPv6
- IPv4 and IPv6 are fully integrated
- IPv6 is enabled and preferred by default

Some IPv6 Security Implications

IPv6 has a number of positive and negative security implications (the following apply in general to IPv6 implementations/installations and hence to Vista environments):

- Doubles (±) the possible attack surface, until IPv4 is dropped
- 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
- And much more

The New Vista Network Stack

- The rewritten Vista stack means there is lots of opportunity for vulnerabilities
  - 1000's of lines of new code
  - Stacks are complex entities that take years to mature
- Microsoft did an extensive security testing and design process
  - This has certainly eliminated many possible vulnerabilities
- In beta 2 builds we found 3 historic stack attacks and 3 crashes from IPv4 fuzzing

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, ICMPv4, ARP, IGMP, etc

Vista IPv6 Transition Mechanisms

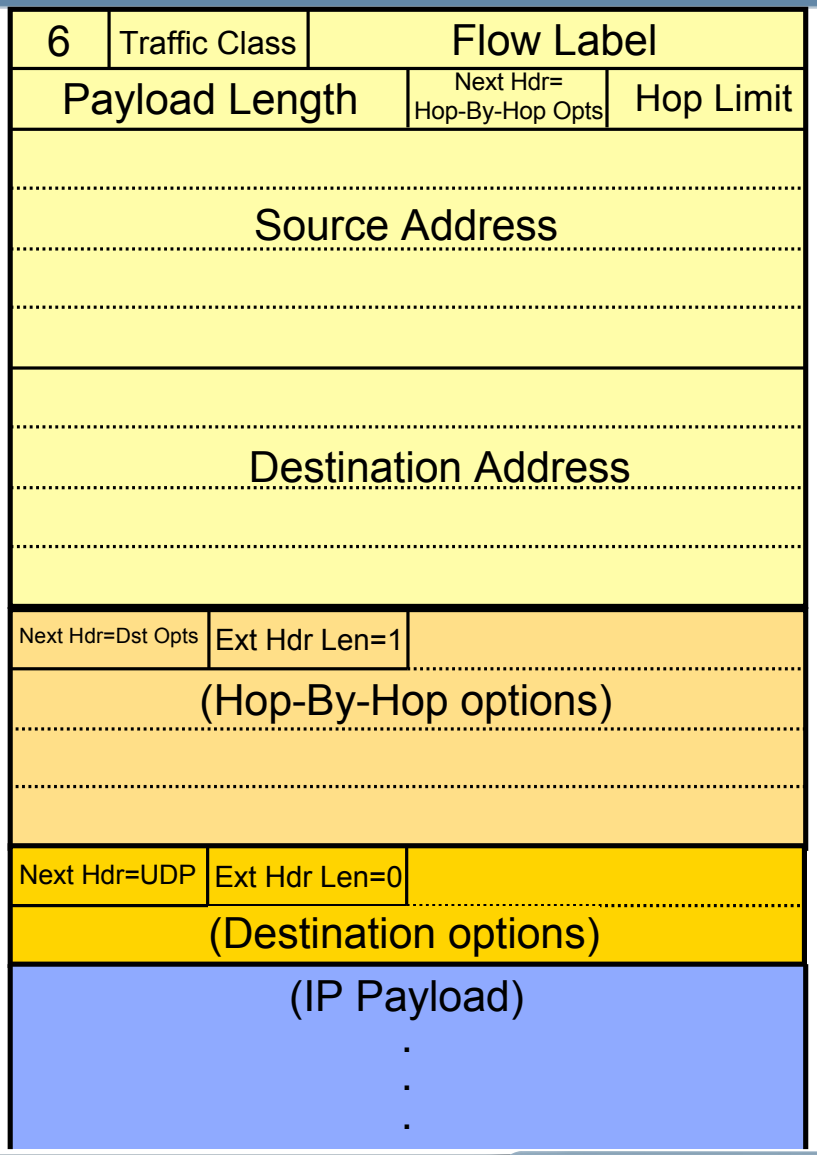
To promote having more clients using IPv6 on the Internet, Microsoft has implemented transition mechanisms for IPv6, including:

- ISATAP
  - IPv6 tunneled directly on top of IPv4
- Teredo
  - IPv6 tunneled on top of UDP over IPv4

These mechanisms can allow hosts to use IPv6 even if something on the path doesn't support it

IPv6 Header

The IPv6 header consists of a simple base header and zero or more extension headers



Defined extension headers include:

- Hop-By-Hop Opts
- Dest. Options
- AH
- ESP
- Fragment
- Routing
- Mobile IPv6*

*: not present in Vista RTM

Windows Firewall for Vista

Vista introduces a new Windows Firewall

- Enabled by default
- Default deny with fairly limited exceptions initially

Vista introduces network profiles

- Every network has an assigned profile
- 3 built-in network profiles
  - Public (default, most restrictive)
  - Private (home or office)
  - Domain (under a domain controller)
- Assigned profile selects the ruleset in place in Windows Firewall

Vista Windows Firewall Incorrectly Applies Filtering to Teredo Interface

- Ollie and I found a vulnerability in Windows Firewall for Vista
- By design:
  - Inbound traffic over Teredo only allowed when the “edge traversal” flag is set in an active firewall exception
  - Out-of-the-box, Windows Firewall allows no inbound traffic
- As implemented (RTM):
  - The firewall allowed over Teredo all inbound traffic that would be allowed from local link
  - Exposure depends on the current firewall rule state
    - For out-of-the-box and TCP, this manifests itself as port 5357 being available over the Teredo interface
- Fixed in MS07-38, documented in SYMSA-2007-05

Windows Firewall State Change Testing

We studied the effect of certain GUI actions in Vista upon Windows Firewall and active sockets

- E.g., enabling file sharing, turning it back off

What we observed:

- Enabling certain features enables 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

GUI action	Firewall sticky exceptions	Profiles
Turn Media Sharing on then off	“Windows Media Player” group	Private and Domain
Sign into People Near Me then quit it	“Windows Peer to Peer Collaboration Foundation” group	All
Sign into Windows Meeting Space then quit it	“Windows Peer to Peer Collaboration Foundation”, “Windows Meeting Space”, and “Network Projector” groups	All
(There are likely others)		

These sticky rules increase the host’s exposure

- Of course, need a listener + a firewall exception for a port to be open
- Sockets usually closely matched GUI state
  - However, TCP port 5722 (DFSR.exe) remained open an extra few minutes after Windows Meeting Space was closed

Outline

1. Introduction
2. Vista's new network stack and firewall
3. Some layer 3 & 4 results
4. IP & TCP reassembly behavior
5. The Teredo protocol
6. Teredo security implications
7. LLTD
8. Conclusion

IPv6 Next Header/IPv4 Protocol Enumeration

Protocols/codes	IPv6	IPv4
Unsupported protocol codes	Produce a param. prob. message, so we can map serviced protos	No such response with firewall on – tested with it off
TCP & UDP	Yes	Yes
ICMPv4		Yes
ICMPv6	Yes	
IGMP		Yes
IPv6 No Next Header	Yes	
ESP & AH	Yes	Yes
Routing/43 & Fragment/44	Yes	**Yes**
Hop-By-Hop & Dest. Opts	Yes	
IPv4 over IPv_	**Only if firewall on**	**Yes**
IPv6 over IPv_	**Yes**	Yes
GRE		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?
  - A different way to do fragmentation or source routing for IPv4?
- Inferring meaning from the lack of a Protocol Unreachable is not necessarily reliable
  - The lack of a negative doesn't establish a positive
  - But does point 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

Available Tunneling in Vista

From IP scans, these tunnels appear to be available:

- IPv4 over IPv4
- IPv4 over IPv6 (needed for IPv4 in an IPv6-only network)
- IPv6 over IPv4 (ISATAP)
- IPv6 over IPv6
- GRE over IPv4 (GRE by design can be used to tunnel any protocol)

(We didn't investigate the actual availability of most of these tunneling mechanisms)

Other tunneling we know of in Vista:

- Teredo
- IPsec tunnel mode
  - Over AH or ESP

Tunneling in Vista

More tunneling is available in Vista than XP

- This is an area of concern due to the possibility of security controls being bypassed

Requirement for a firewall:

- On Vista, the Teredo component refuses to start up unless an IPv6 firewall is in place
- There may be the same safety check for IPv4 over IPv6
  - Since there was a protocol unreachable only when the firewall was off

TCP Port Enumeration

Scanning from the 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)

Same result for scanning a Teredo interface from the Internet (prior to Windows Firewall fix MS07-038)

UDP Port Enumeration

Scanning from the 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/Link Local Multicast Name Resolution

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, we had to slow down our IP proto and UDP port scanning
  - Since those depend on ICMP error messages
  - 18 hours for a simple UDP port scan
- This slows down legitimate and malicious scanners
  - Unless they work around it (e.g., using multiple sources)

Miscellaneous Vista Layer 3&4 Observations

- By default, Vista does not respond to pings
- Vista only uses half the available IPv4 ID range (0 to 0x7FFF)
  - It uses the range sequentially
  - Should still be able to do host counting behind a NAT
- Ephemeral port range has changed
  - Now 49152 to 65535
- TCP ISN generation seems good

Outline

1. Introduction
2. Vista's new network stack and firewall
3. Some layer 3 & 4 results
4. IP & TCP reassembly behavior
5. The Teredo protocol
6. Teredo security implications
7. LLTD
8. Conclusion

NIDS Evasion With Fragments

- It is possible (though not legitimate) to send an ambiguous sequence of IP fragments or TCP segments
  - E.g., different data sent for same part of packet
- Different TCP/IP stacks will interpret these in different ways
  - Neither the TCP, IPv4, or the IPv6 specifications say how to treat these
- This creates a challenge for network-based IDS/IPS since it needs to predict and match the recipient system's behavior
  - Otherwise face evasion
- See *Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection* by Thomas Ptacek and Tim Newsham

TCP Segment Reassembly

Example:
four overlapping TCP segments:

```
  is_is
      _bad
    at_m
That
```

Vista: `This_is_bad`
XP: `That_is_bad`
Linux: `That_is_mad`

- We empirically studied Vista's TCP segment reassembly behavior
- It is different than XP or other stacks
  - Old data is always preferred over newer data
- IDSs will have to adapt to prevent evasion attacks

Vista's IP Fragment Reassembly

- We 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
- This means IDSs will have to adapt to prevent evasion attacks

IP Fragment Reassembly (Full Overlap)

- Two fully overlapping fragments

```
        AAAAAAAA
        BBBBBBBB
CCCCCCCC
```

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

```
CCCCCCCCAAAAAAAA
```

- Linux: favor new

```
CCCCCCCCBBBBBBBB
```

IP Fragment Reassembly (Partial Overlap, General Case)

- Two partially overlapping fragments

```
                BBBBBBBBBBBBBBBB

AAAAAAAAAAAAAAAA
```

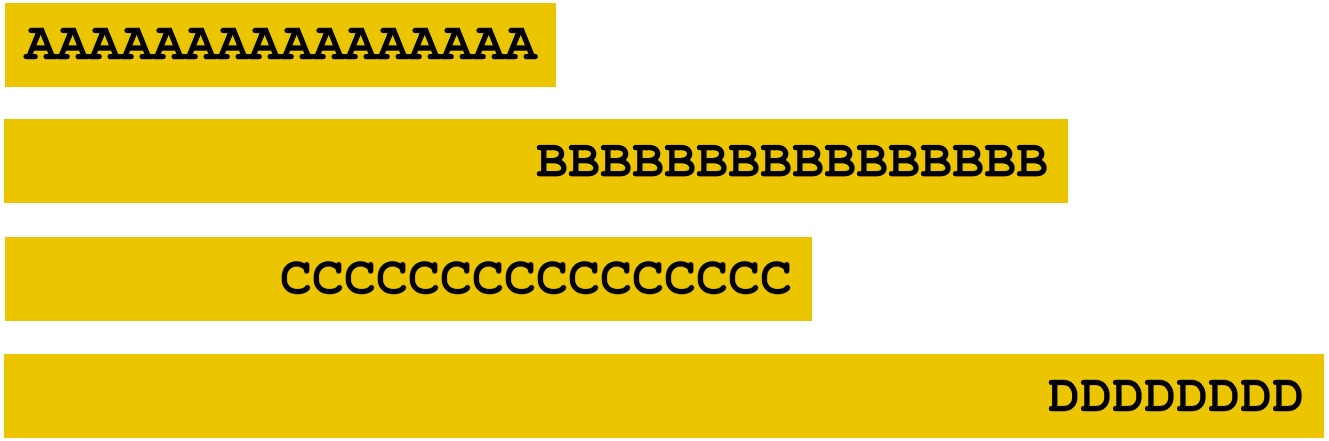
- XP: prefer previous data (favor old)

```
AAAAAAAABBBBBBBBBBBBBBBB
```

- Vista: packet not reassembled

IP Fragment Reassembly (Overlap Within Leading Range)

- 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



- Reassembled:

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

Observing IPv4 Fragment Reassembly

Somehow, we need to observe how the packet is assembled

IPv4 reassembly testing:

- The region that is fragmented ambiguously is the payload of a UDP packet
- Run netcat on the recipient system such that the system's stack will pass the reassembled packet to it (nc -u -l)
- 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 reassembly testing:

- We use the approach of sending a packet that, when reassembled, will yield an ICMPv6 error
  - I.e., we intentionally cause an error after reassembly completes
  - We receive the error (including the “original” packet), so we can see how the packet 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 be sent if it is not understood
- Approach takes advantage of a new requirement with IPv6:
  - The *full* original packet must be included in an ICMPv6 error message (up to 1280 octets in return packet)

Outline

1. Introduction
2. Vista's new network stack and firewall
3. Some layer 3 & 4 results
4. IP & TCP reassembly behavior
5. The Teredo protocol
6. Teredo security implications
7. LLTD
8. Conclusion

Teredo Introduction

Teredo was developed by Christian Huitema of Microsoft

- Published as RFC 4380 (“Teredo: Tunneling IPv6 over UDP through NATs”)
- Standards track individual submission

Teredo functional niceties:

- Works through NATs and with hosts possessing no public addresses
- Automatic tunnel setup
- Teredo client is provided with a global IPv6 address and is globally addressable
- No support needed from local network (public Teredo servers and relays are used)
- Peer IPv6 host need not be aware of Teredo
- Local applications need not be aware of Teredo

The Use For Teredo

Why is Teredo often needed for IPv6 connectivity?

- Many of the computers on the Internet are behind IPv4 NATs
- IPv4 NATs don't support native IPv6 or even ISATAP
- NATs (especially home gateways) are rarely upgraded

However, Teredo is only supposed to be used when native IPv6 and ISATAP are not available

- Since it is less efficient and less reliable
- An IPv6 provider of last resort [RFC 4380]

Teredo Implementations

Vista:

- Teredo is enabled by default on Windows Vista
- It is the IPv6 provider of last resort
- But may be preferred over native IPv4 in some situations
- May often get used in Vista
  - Microsoft hasn't clearly documented the cases in which it will be used
  - We saw it used more frequently than the MS documentation initially said
- Safest to assume Teredo will often be in use for Vista hosts

Windows XP SP2 and Windows Server 2003 SP1:

- Teredo available but disabled by default

Unix and Mac:

- Open source Teredo implementations are available (e.g., Miredo)

Teredo Component: Teredo Server

- Teredo servers are their client's helpful friends with the right connections
  - That is, they have native IPv6 access
- Teredo servers help the client set up its Teredo address
  - Server reports back to client what its external IP address and port is
  - Server also helps client determine if its NAT is compatible with Teredo
- The server for a client to use is usually statically configured
  - This is the only part of Teredo that is not entirely automatic
  - Vista: out of box configured to use teredo.ipv6.microsoft.com (resolves to 9 IPs)

Teredo Addresses

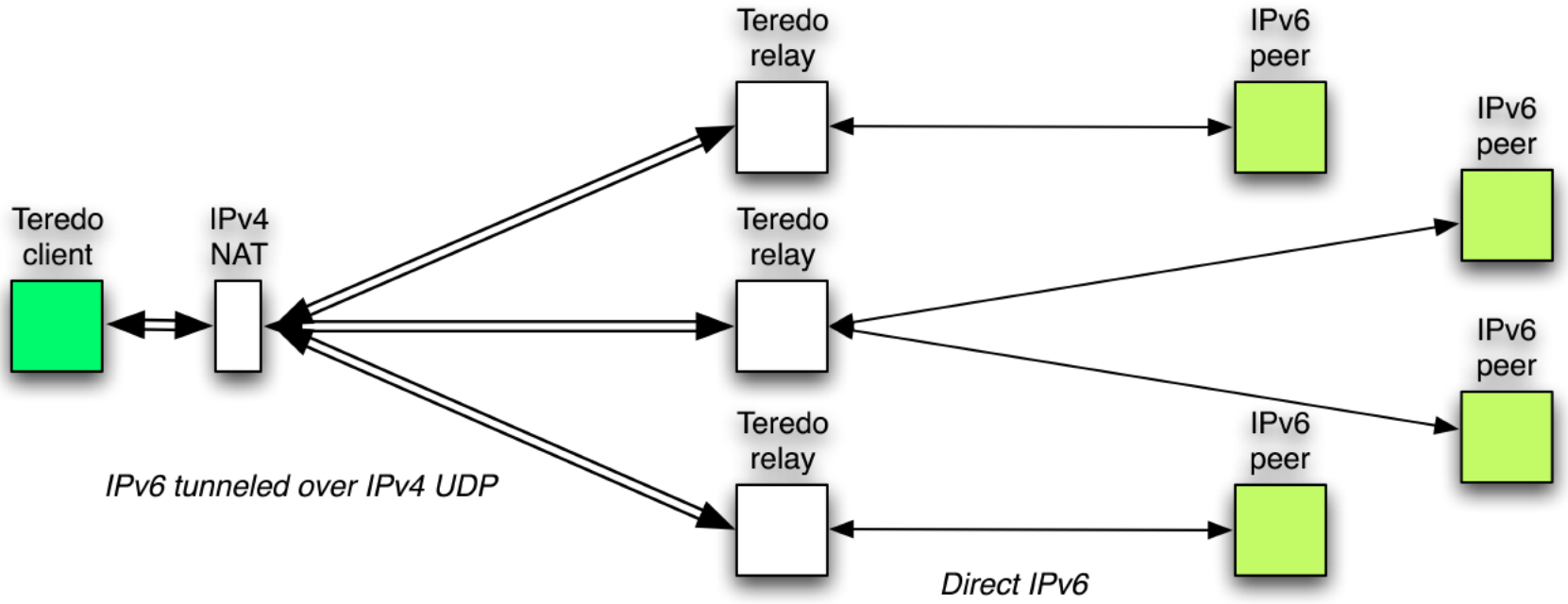
Teredo address format (128 bits):

Teredo prefix	
Server IPv4 address	
Flags	Client port # (bit-flipped)
Client IPv4 address (bit-flipped)	

- `2001:0000::/32` is the assigned address prefix
- These addresses are unique and have global scope (are globally routable)

Example: `2001:0:4136:e37a:0:1080:f580:ea94`

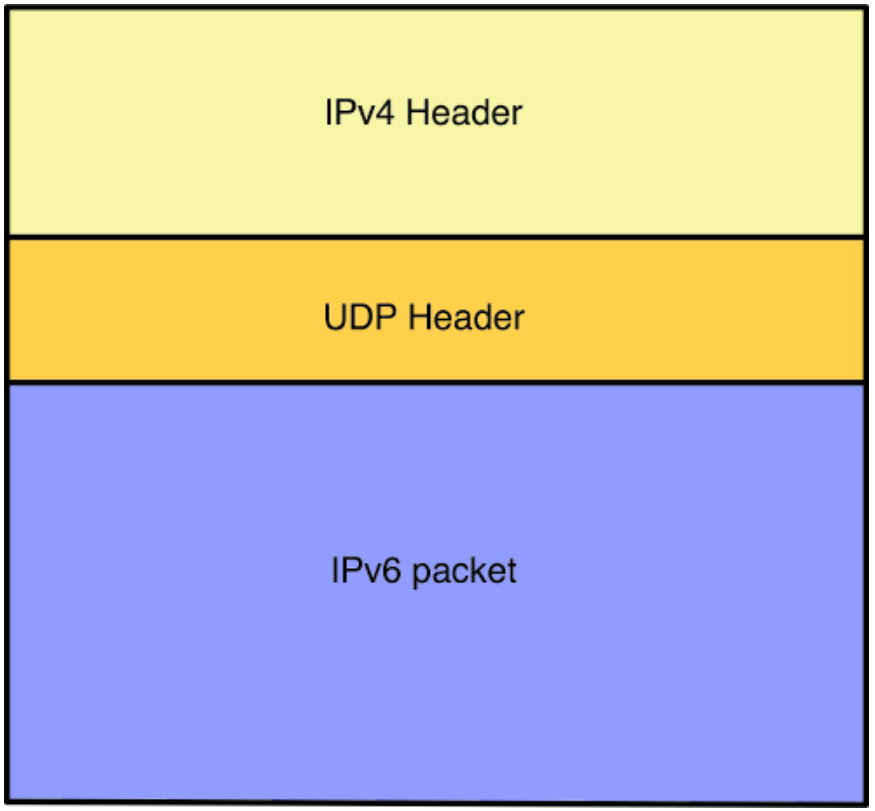
Teredo Component: Teredo Relay

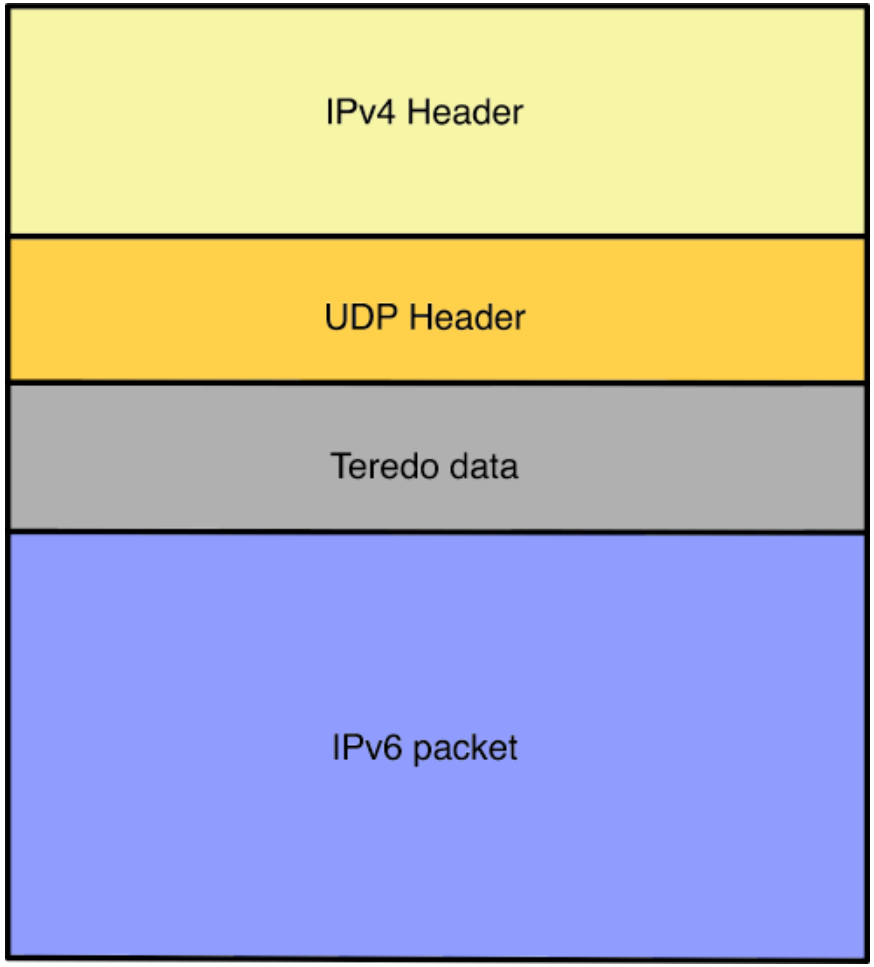


Using a relay, both Teredo clients and peers can initiate a packet send

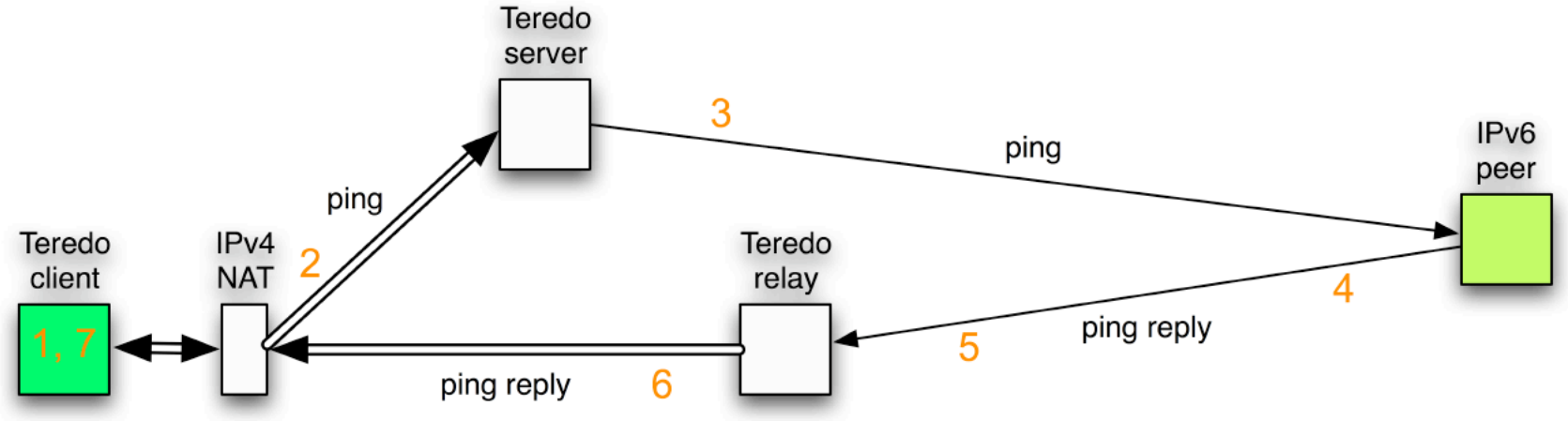
- Native IPv6 peer finds closest relay since relays advertise a route to `2001:0000://32`
  - Teredo addresses contain enough information for a relay to reach a Teredo client by IPv4
- Teredo client finds a relay to use with help from Teredo server
  - Ping test establishes what relay will be used to reach a peer
  - Also used to guard against peer spoofing

Teredo Encapsulation





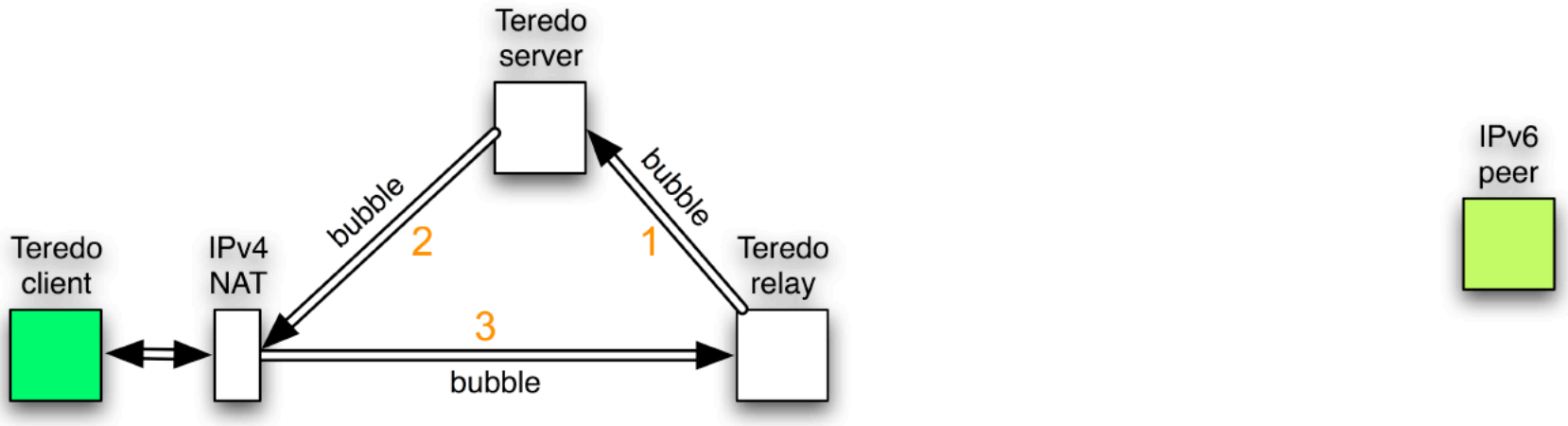
Ping Test Procedure (Used For Each New Peer)



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 the ping and drops it on the IPv6 Internet
4. Peer responds to ping as normal
5. Echo reply is routed to nearest relay
6. Relay encapsulates this and passes it to client via IPv4
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 the client preventing timeout

Outline

1. Introduction
2. Vista's new network stack and firewall
3. Some layer 3 & 4 results
4. IP & TCP reassembly behavior
5. The Teredo protocol
6. Teredo security implications
7. LLTD
8. Conclusion

Teredo Security Implications

- Teredo raises security concerns (some serious), including:
  - Unexpected host accessibility
  - Network security control bypass
  - Cost to locate Teredo IPv6 payload
  - Peer address disclosure
  - Pharming/phishing with Teredo
  - (several more in the paper)
- Also provides a few security positives
- Some of the concerns are not mentioned in RFC 4380
- I submitted draft-hoagland-v6ops-teredosecconcerns-00 to the IETF to document our concerns
  - Now maintained by Suresh Krishnan (Ericsson) and taken up as a v6ops working group Internet Draft

Security Concern: Host Accessibility

- Teredo puts hosts directly on the Internet
  - Teredo addresses are global addresses
  - Anyone can send an IPv6 packet directly to Teredo client
  - Provides a stable open-ended tunnel
- End-to-end connectivity is the way it is supposed to be with IPv6
  - However, with native IPv6, admins would be aware of the exposure
  - With Teredo, hosts will be unexpectedly exposed
    - Even if they only have a private IPv4 address and are behind a NAT
- Vista:
  - Teredo may often be active
  - Windows Firewall default denies all inbound Teredo packets (after MS07-038)
  - Vista does require an IPv6 capable firewall to be registered

Security Concern: Teredo Bypassing Security Controls

- Teredo's IPv6 content bypasses inspection by network security components (e.g., firewall, network IPS)
  - … unless they are specifically Teredo aware
- This means network controls won't be applied
  - Some important controls may not be in place on end-host
  - Defense in depth is reduced in any case
- Those defenses were in place for a reason, right?
- You should be applying at least as strong controls to Teredo tunnel packets as to IPv6

Security Concern: Cost To Find All Teredo Packets

Inspecting all Teredo content (selective filtering, passive monitoring):

- Inspecting the IPv6 content of Teredo packets on the wire is not trivial
  - Only server-bound traffic has a characteristic port (UDP 3544)
  - So, need to apply a heuristic to all packets on all UDP ports
    - Can be expensive
- In some situations, this may make it infeasible to do security inspection of the Teredo tunneled content on the network

Blocking all Teredo:

- Blocking outbound port 3544 should *eventually* starve normal Teredo clients of ability to connect by blocking access to server
  - Especially if applied before the NAT
  - Will not prevent outbound malicious or intentionally evasive connections though

Recent IETF v6ops Direction on Teredo

- The difficulty in inspecting Teredo tunneled IPv6 packets has recently seemed to motivate the IETF v6ops working group towards consensus that Teredo should not be used in managed networks
- Internet Draft draft-ietf-v6ops-teredo-security-concerns-00:
  - "Teredo is NOT RECOMMENDED as a solution for managed networks."
  - http://ietfreport.isoc.org/idref/draft-hoagland-v6ops-teredosecconcerns/
- Christian Huitema (Teredo author):
  - "If an organization wants to provide IPv6 connectivity while monitoring the IPv6 traffic, then Teredo is definitely not the right tool. … The best way to achieve that is to provide native IPv6 connectivity. If the organization's internal network cannot be upgraded to support native IPv6, then it should consider other transition technologies like ISATAP, rather than Teredo."
  - http://ops.ietf.org/lists/v6ops/v6ops.2007/msg00459.html

Teredo Security Positives

- RFC 4380 requires a lot of sanity checking on packets
  - Prevents a number of attacks
  - Have verified that Vista does at least some of them
- Can use IPsec in normal manner
  - Hard to use IPsec with 6to4
- Teredo specifies decent anti-spoofing mechanisms to be used (e.g., ping test)
  - Beneficial for case where IPsec is not being used
  - Vista (as of RC2):
    - Ping test nonce is only 32 bits (RFC suggests at least 64 bits)
    - Also, sometimes “0” is used as nonce instead of random

Security Concern: Peer Address Disclosure

- Server knows (essentially) all of a client's peer IPv6 addresses
- This is since helps with ping test
- Okay if you trust the server not to make bad use of it
- Vista and XP: use Microsoft servers by default
  - Any conspiracy theorists out there?
  - Can probably trust Microsoft on this. Right?

Security Concern: Teredo Server Bumping (1)

What if some malware or malicous user changes 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 (by responding to the ping test itself)
  - 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 client prefers Teredo over native IPv4
- Potential for the server to spoof all IPv6 capable servers (or other peers) on Internet

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 it is easier to miss a bumped server
  - Also it always says that Teredo is not being used

Teredo Suggestions

Due to the security implications we've found, for managed networks I recommend:

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

Outline

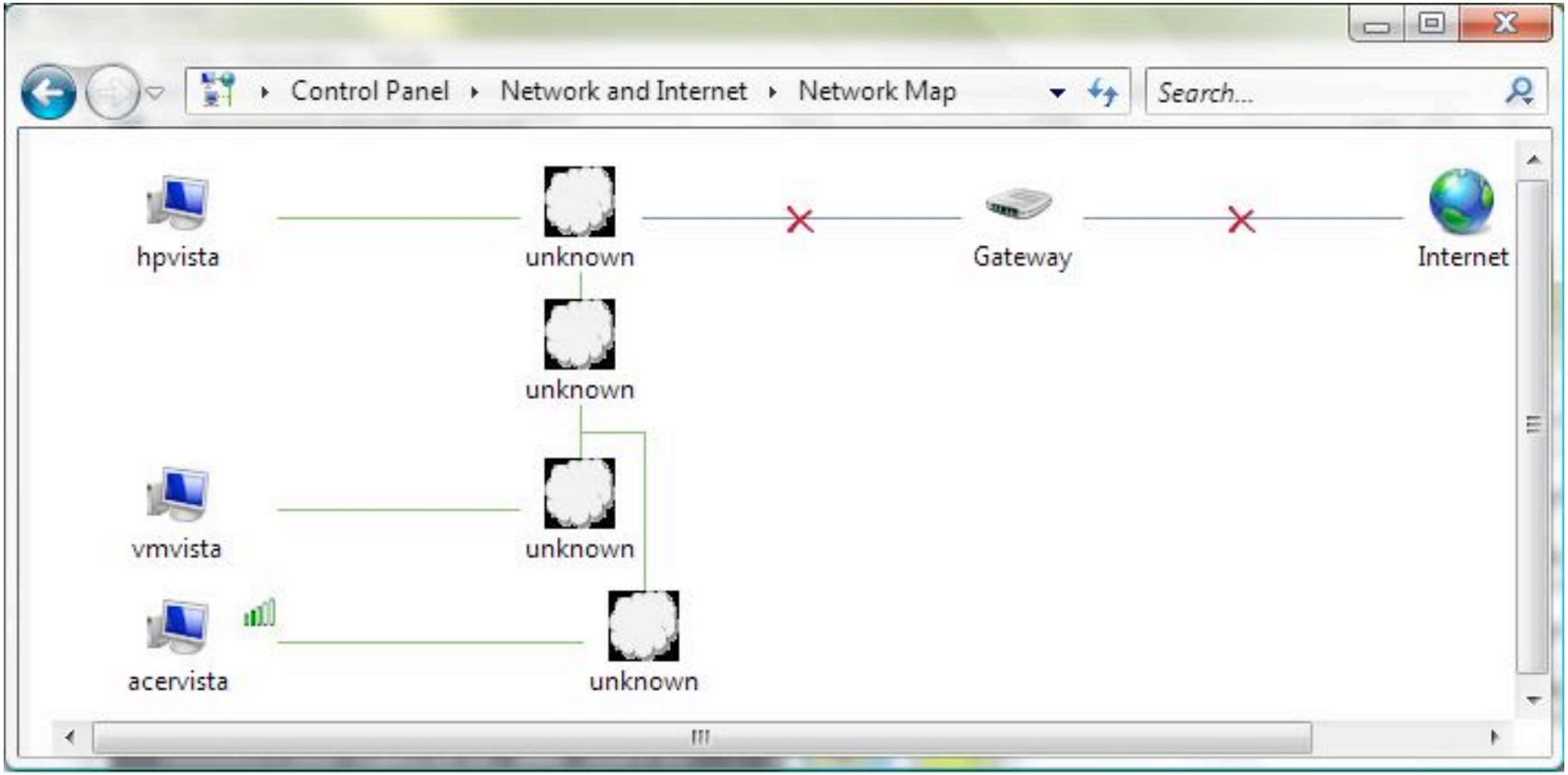
1. Introduction
2. Vista's new network stack and firewall
3. Some layer 3 & 4 results
4. IP & TCP reassembly behavior
5. The Teredo protocol
6. Teredo security implications
7. LLTD
8. Conclusion

LLTD Research

- We looked into the Link Local Topology Discovery (LLTD) protocol and Vista's implementation of it
- Performed on beta 2 build 5472 (July '06, results 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



- Protocol runs directly over Ethernet
- Documented:
  - http://www.microsoft.com/whdc/Rally/LLTD-spec.mspx

LLTD Research Conclusions

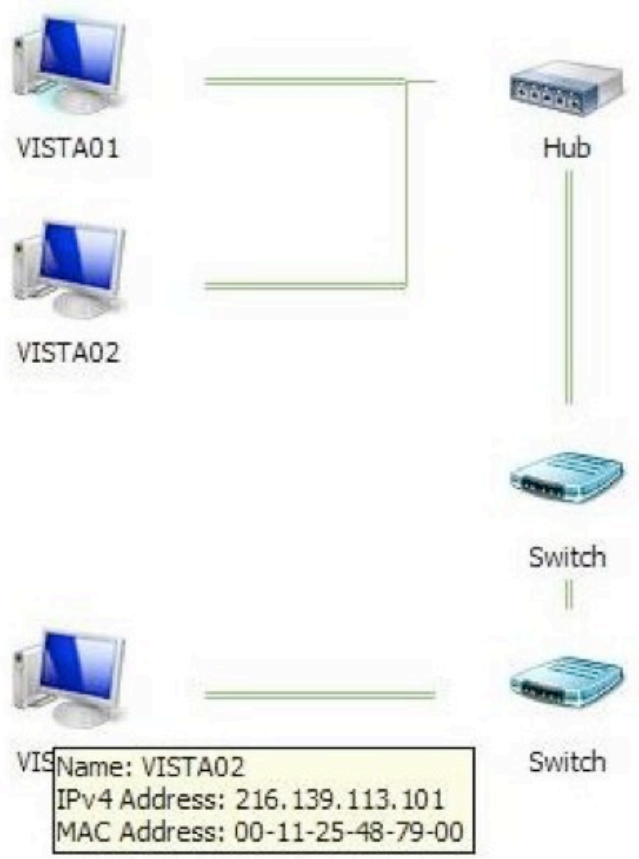
Conclusions:

- 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 Security Development Lifecycle throughout the protocol design and implementation

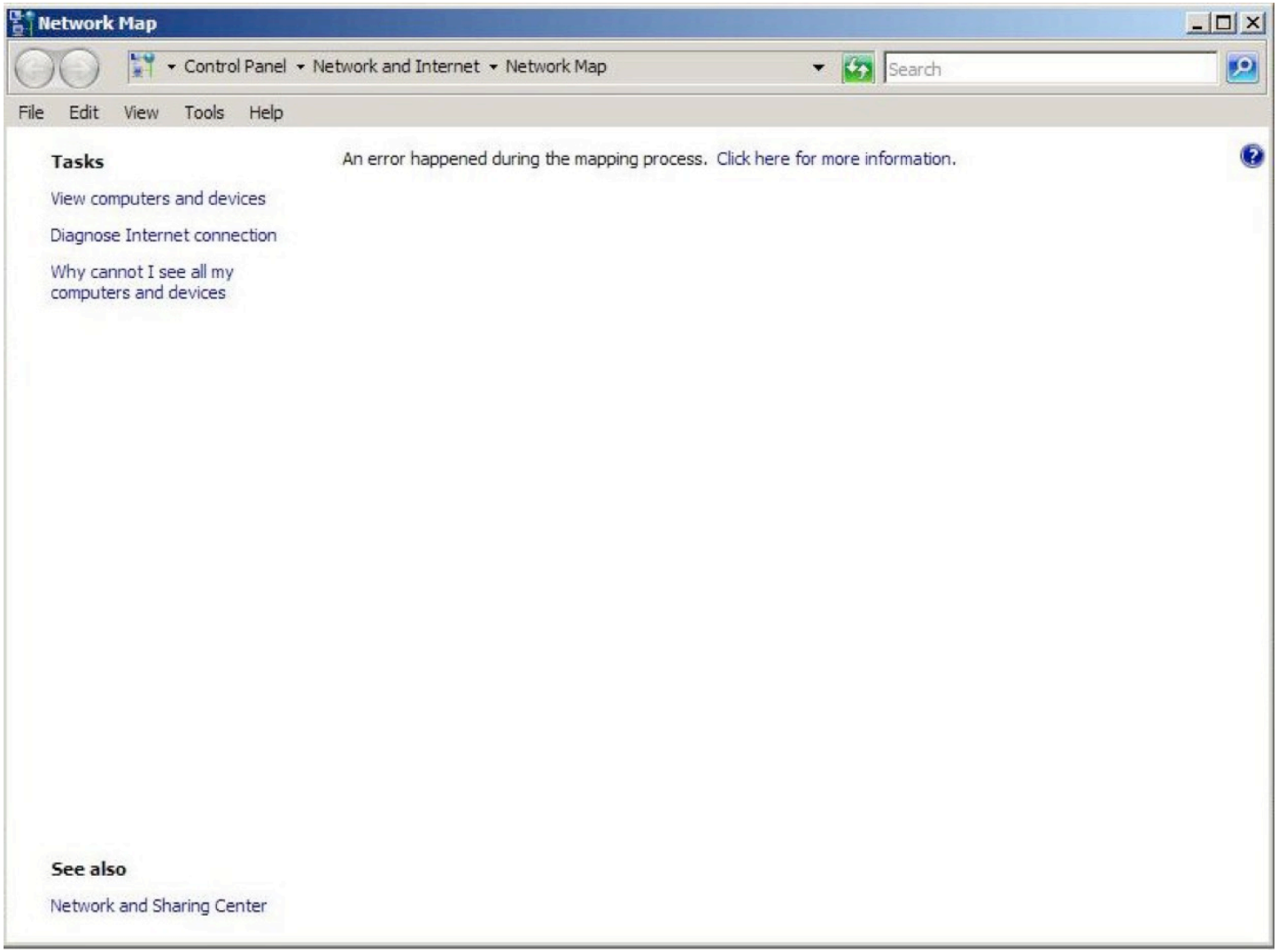
LLTD doesn't raise many concerns, however:

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

Example of Faking Data on Network Map Using LLTD



DoS of Network Mapping with Malicious LLTD Responder



Outline

1. Introduction
2. Vista's new network stack and firewall
3. Some layer 3 & 4 results
4. IP & TCP reassembly behavior
5. The Teredo protocol
6. Teredo security implications
7. LLTD
8. Conclusion

Conclusion

- Beware of Teredo tunneling through your network
  - It may be imitating what the namesake mollusk does to ships
- See what you need to do as a result of the networking changes in Vista

- Read our reports for more details
  - http://www.symantec.com/avcenter/reference/Vista_Network_Attack_Surface_RTM.pdf
  - http://www.symantec.com/avcenter/reference/Teredo_Security.pdf

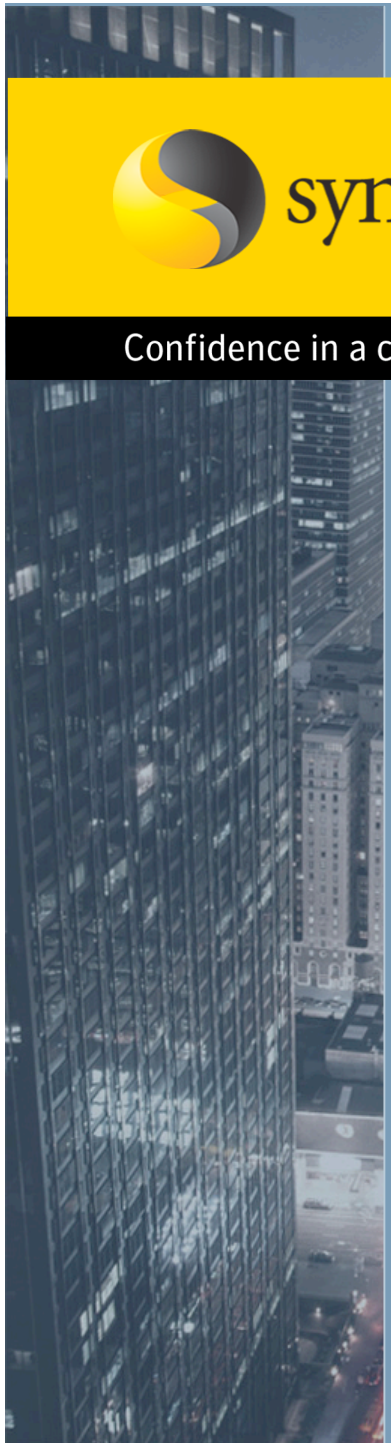
Questions?

Thank you!

Jim Hoagland

jim_hoagland@symantec.com

http://www.symantec.com



symantec™

Confidence in a connected world.

Bonus Slides

Symantec ATR Vista Reports

All ATR reports available via http://www.symantec.com/enterprise/security_response/weblog/

- *Windows Vista Network Attack Surface Analysis: A Broad Overview*
  - By Tim Newsham and Jim Hoagland
- *Analysis of the Windows Vista Security Model*
  - By Matt Conover
- *Assessment of Windows Vista Kernel-Mode Security*
  - By Matt Conover

RTM reports also available at: http://www.symantec.com/enterprise/theme.jsp?themeid=vista_research

- *Security Implications of Windows Vista*
  - By Oliver Friedrichs and Ollie Whitehouse
- *The Impact of Malicious Code on Windows Vista*
  - By Orlando Padilla
- *Analysis of GS Protections in Windows Vista*
  - By Ollie Whitehouse
- *An Analysis of Address Space Layout Randomization on Windows Vista*
  - By Ollie Whitehouse
- Plus the two being presented today

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 (undefined)
  - Length = 0 (illegal, should be ≥2)
- Beta 2 build 5270: Target became 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):

- Land
  - SYN with source IP=destination IP
  - Attempt to cause host to reply to itself
  - Network stack was unresponsive for a few seconds
- Blat
  - SYN flood with URG pointer pointing past end of packet
  - 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

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)

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)

Default Source Routing Behavior on Vista

- Source routing is an IPv4/IPv6 feature where the packet originator specifies the routing path
- Based on netsh examination and empirical testing:

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

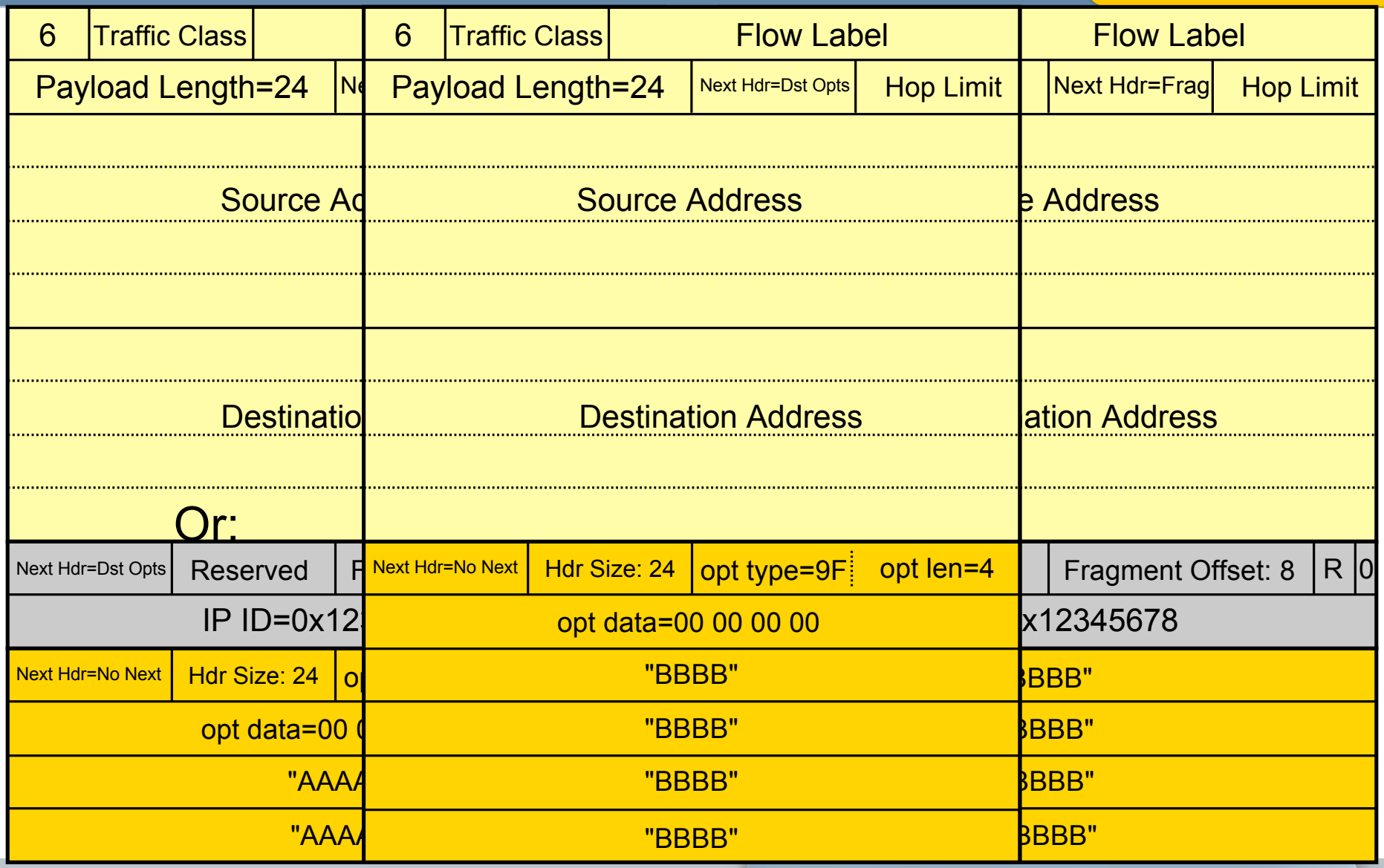
- Routing Header type 0 (RH0) recently a concern for IPv6
  - Vista accepts at end but does not forward

ARP and ND Attacks

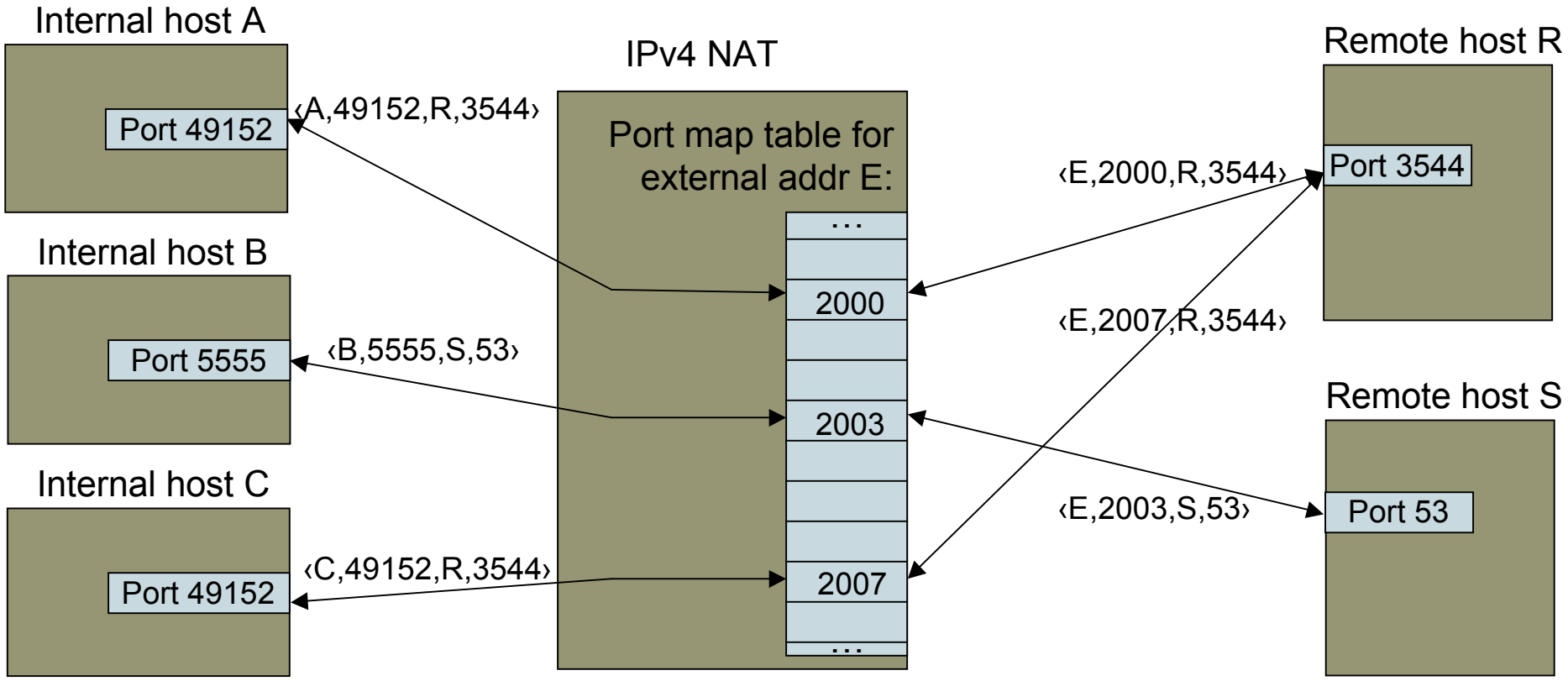
- Attacker can cause false IPv4/6-MAC assoc. in some cases
  - A.k.a. cache poisoning (enables man-in-the-middle, DOS)

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 ARP table entry and gets used	Creates neighbor cache entry and gets used
Solicited false reply for address with no entry (broadcast/multicast reply)	Not stored but will be used if needed	Creates neighbor cache entry and gets used
Faked address conflict	Statically configured addr.: like XP, interface becomes unusable until reset	Link-local RFC 3041 address: automatically generates new address

Assembling IPv6 Fragments



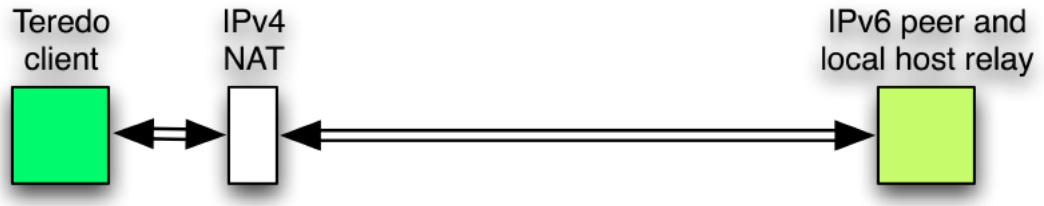
Review: What Do NATs Do?



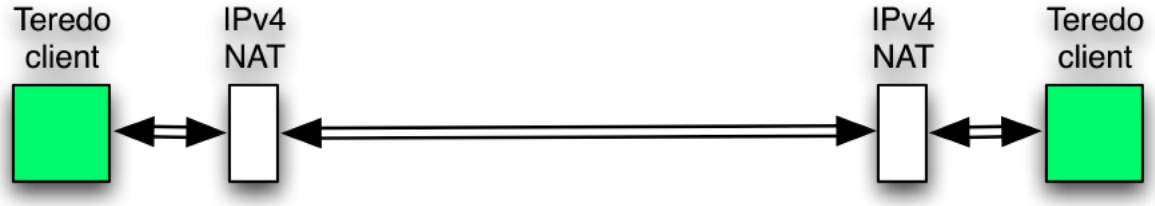
- NATs map a port number on an internal host to a port number on a public IP address
- Thus they provide an Internet presence for the host's internal port
  - Some NATs filter inbound traffic to only allow packets from recent outbound peers

May Not Need an Internet-based Teredo Relay

- If IPv6 peer has both global IPv6 and IPv4 addresses and is Teredo-aware, it can be its own “local host relay”
  - The packet is encapsulated before leaving peer
  - Thus it is tunneled for full route (no IPv6 networks needed)
  - Vista and Windows Server 2008: 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 + Source Routing

What if a 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

Could also use source routing to sent packets through a specific Teredo relay

- A way to choose a relay other than the normal one
- Could be used as part of an attack

Security Concern: Teredo Information Disclosure (Teredo Address Data)

There are some fields in the Teredo address that can reveal useful information to an attacker, including…

- Cone bit (in flags field)
  - Cone bit in Teredo address left unset means client's NAT isn't restrictive in terms of who is routed in
  - A sign of weakness
- Server field
  - If it is a Microsoft address, host is probably a Windows host
  - Could be used to target attacks or profile targets

Security Concern: Denial of Teredo Service

- There are various ways to kill or degrade Teredo service at a client or relay
  - Relay would affect multiple hosts
- Maybe even at a server

Ways to Find a Teredo Host

- Have the Teredo host connect to you (run a IPv6 web server and try to get connections)
- See the address on P2P, in a log file, on the wire, or at a Teredo component
- Scan Teredo addresses (may be feasible, especially when focused on a particular target)
- More?

Security Concern: Teredo Address Scanning (1)

- 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 addresses 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

Security Concern: Teredo Address Scanning (2)

- Vista adds in 12 random bits in address (flags field)
  - Not mentioned in Teredo RFC
  - This makes addresses 4096 times harder to guess
  - Note: actual randomness of the 12 bits hasn't been studied
- Vista clients:
  - Server field pretty predictable
  - Client port number drawn from 49152-65536
    - Will sometimes make external port number more predictable

Security Concern: Teredo and Worms

- Main benefit to worms 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
  - E.g. a vulnerability in IPv4 option processing
  - Might be able to spread with a single UDP packet like Slammer