IPv6 Operations Working Group
Internet-Draft
Expires: January 27, 2008

J. Hoagland
Symantec
S. Krishnan
Ericsson

July 26, 2007

Teredo Security Concerns
draft-ietf-v6ops-teredo-security-concerns-00

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on January 27, 2008.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

Additional security concerns with Teredo are documented, beyond what is in RFC 4380. This is based on an independent analysis of Teredo's security implications. The primary intent of this document is to provide information and recommendations to the IETF that can be used in any updated Teredo specification. The second intended audience is anyone that can help improve security in Teredo as deployed, so they will be aware of these concerns.

Table of Contents

1. Introduction

An independent analysis of Teredo's security implications was conducted by Symantec[TTPTPNSOSI], based on the Teredo specification ([RFC4380]). This analysis uncovered some security concerns associated with Teredo which are not documented in the Teredo specification. This document discloses these additional concerns and includes any recommendations where relevant. This Internet Draft is also influenced to an extent by an examination of the Teredo implementation on Microsoft Windows Vista [WVNASA].

The primary intent of this document is to provide information so that can be used in any updated Teredo specification. Secondarily, this document can help improve security in Teredo as deployed (including those that implement Teredo, security providers, and network security administrators) become aware of any valid security concerns.

2. Teredo Bypasses Security

2.1. Teredo Bypasses Network Security

2.1.1. Problem

IPv6 traffic tunneled with Teredo will not receive the intended level of inspection or policy application by network-based security devices, unless the devices are specifically Teredo aware and capable. This reduces defense in depth and may cause security gaps. This applies to all network-located devices and to end-host based firewalls whose existing hooking mechanism(s) would not show them the IP packet stream after the Teredo client does decapsulation.

2.1.2. Discussion

Evasion by tunneling is often a problem for network-based security devices such as firewalls, intrusion detection and prevention systems, and router controls. The vendor of such devices must add support for detunneling for each new protocol. There is typically a significant lag between when the vendor recognizes that a tunnel will be used (or will be remotely usable) to a significant degree and when the detunneling can be implemented in a product update, the update tested and released, and the customer begins using the update. Late changes in the protocol specification or in the way it is implemented can cause additional delays. This becomes a significant security concern when a delay in applied coverage is occurring frequently.

Specifically for Teredo, a Teredo-unaware network security device would inspect or regulate the IPv4 and the IPv4-based UDP layer as

normal for IPv4, but it would not recognize that there is an additional IP layer contained inside the UDP payload that it needs to apply the same controls as it would to a native packet. (Of course, if device discards the packet due to something in the IPv4 or UDP header, such as referring to an unknown protocol, the Teredo packet is no longer a concern.) Teredo also only recently reached RFC status (February 2006), is widely applicable, requires no support from the local or organizational network, and looks ready to be widely used. Furthermore the tunnel created by the Teredo client is open-ended and allows bidirectional traffic.

Network security controls being not applied must be a concern to those that set them up, since those controls are supposed to adequately regulate all traffic. If network controls are being bypassed due to the use of IPv6 via Teredo, the burden of controls shifts to the Teredo client host. Since security administrators may not have full control over all the nodes on their network, they sometimes prefer to implement security controls on the network.

One implication of the security control bypass is that defense in depth has been reduced, perhaps down to zero unless a 'local firewall' is in use, as recommended as a mitigation in RFC 4380. However, even if there are host-based security controls that recognize Teredo, security administrators may not have configured them with full security control parity, even if all controls that were maintained by the network are available on the host. Thus there may be gaps in desired coverage.

Compounding this is that, unlike what would be the case for native IPv6, some network administrators will not even be aware that their hosts are globally addressable; for example, they may not be expecting this for hosts with RFC-1918 [RFC1918] addresses behind a NAT. In addition, Section 3.2 discusses how it may not be efficient to find all Teredo traffic for network devices to examine.

2.1.3. Recommendations

Of course security administrators should disable Teredo functionality unless their network-based security controls adequately recognize the tunneled traffic (unless they consider it an acceptable risk). However, there may be an awareness gap. Thus, due to the possible negative security consequences, we recommend that explicit user action be required to enable a Teredo client for the first time, at least for the time being. When Teredo is being enabled or when it is going to be used for the first time, perhaps there should be a descriptive warning about the possible evasion that will occur. In addition, Teredo client functionality should be easy to disable on the host and through a central management facility if one is

provided.

RFC 4380 requires that Teredo be an IPv6 provider of last resort. To minimize security exposure due to Teredo, we recommend that Teredo also be an IP provider of last resort. Specifically, we suggest that when both IPv4- and IPv6-based access to a remote host is available, that the IPv4-based access be used in preference to IPv6 access that needs to use Teredo. This should also promote greater efficiency and reliability.

We specifically note that we could find no pre-existing mechanism for Teredo to use that could automate its functionality being disabled unless all network-based security controls were aware of it. A separate type of consent request packet would be needed. (Such a consent request service could have application beyond Teredo.)

2.2. IPv6 Ingress and Egress Filtering Bypass

2.2.1. Problem

IPv6 addresses inside Teredo tunnels are not subject ingress and egress filtering, unless extraordinary measures are taken.

2.2.2. Discussion

Ingress filtering (sanity-checking incoming destination addresses) and egress filtering (sanity-checking outgoing source addresses) are done to mitigate attacks and to make it easier to identify the source of a packet and are considered to be a good practice. This is most naturally (and in the general case, by requirement) done at network boundaries. Teredo-tunneled IPv6 traffic bypassing this network control is a specific case of Section Section 2.1, but is illustrative.

2.2.3. Recommendations

The recommendations in Section 2.1.3 can help here. For this problem specifically, there are two locations in which ingress and egress filtering could be restored.

Network based: network-based devices (e.g. routers) could be updated to find all Teredo packets and to apply ingress and egress controls equally to Teredo tunneled IPv6-addresses.

Teredo client based: Teredo clients could make an effort to conduct ingress and egress filtering. However, there are at least two problems inherent in attempting to do address filtering from this vantage point: knowing the network addresses to filter (drop the

packets of) and knowing whether a peer is from the same network.

The network addresses to filter could be approximated from enumerating the addresses on the network interface the Teredo client is using; at least the /64 of global unicast addresses can be assumed to be in use on the network. Router Solicitations [RFC2461] could also be made.

Peers known to be local due to the Teredo local discovery procedure can be excluded from filtering, but the scope of that knowledge is limited to a broadcast domain, whereas ingress and egress filtering generally applies to a larger scope.

2.3. Source Routing After the Teredo Client

2.3.1. Problem

If the encapsulated IPv6 packet specifies source routing beyond the recipient Teredo client, the host may forward the IPv6 packet to the specified next hop. This may be unexpected and contrary to administrator wishes and may have bypassed network-based source routing controls.

2.3.2. Discussion

IPv6 source routing, while provided for in RFC 2460 [RFC2460] and required in some cases such as mobile IPv6, is often not needed or desired in a given network. Thus it is often blocked, at least for certain types of source routing. The danger is that source routing, by providing a reflection point, violates assumptions made in network security decisions. In addition there is often no compelling case for why it would be needed.

Consider the case where a Teredo packet reaches a Teredo client (and is accepted) and the encapsulated IPv6 packet contains a Routing header. The Routing header indicates that this is not the intended destination (just a hop in a source route). Unless the Teredo client has source routing disabled, it would pass the IPv6 packet on to the next hop. One way to use this for an attack is to have the next hop be a node internal to the network the client is on. Another is to pass the packet back outside, using the Teredo node as a reflection point.

This behavior is not specific to Teredo packets; it works in the same way for all IPv6 packets. However, with native IPv6 packets, a gateway prohibition of source routed packets would have prevented the packet from even reaching the internal host. With Teredo active, the burden is placed upon the end hosts, at least those running Teredo.

Source routing post-Teredo may also be a surprising possibility (packets on an end-to-end tunnel not stopping at the end) that might not have been anticipated in network controls, especially given that a NAT was traversed in the process. RFC 4380 made no mention of source routing.

2.3.3. Recommendations

Teredo clients should by default discard tunneled IPv6 packets that specify additional routing, though they may also allow the user to configure what source routing types are allowed. All pre-existing source routing controls should be upgraded to apply these controls to Teredo tunneled IPv6 packets as well.

3. Challenges in Inspecting and Filtering Content of Teredo Data Packets

3.1. Inefficiency of Selective Network Filtering of All Teredo Packets

3.1.1. Problem

There is no mechanism to both efficiently and immediately filter all Teredo packets. This limits the ability to prevent Teredo use on a network.

3.1.2. Discussion

Given concerns about Teredo security or a network's lack of preparedness for Teredo, a network administrator may wish to simply block all Teredo use. He or she may wish to do so using network controls; this could be either due to not having confidence in the ability to disable it on all hosts attached to the network or due to wanting an extra layer of prevention.

One simple method to do that is easy to employ is to block outbound packets to UDP port 3544. This prevents a Teredo client from connecting to its server and completing qualification. Thus it can be assured that a host trying to establish a new Teredo address will be prevented from using Teredo tunneling. However, existing Teredo clients will not be affected, at least not immediately. In addition, if the blocking is applied on the outside of client's NAT, the NAT will retain the port mapping for the client and the client may or may not continue to use its Teredo address. It is not known if blocking all outbound port 3544 will interfere with non-Teredo traffic.

The other approach is to find all packets to block in the same way as would be done for inspecting all packets (Section 3.2). However,

this faces the difficulties in terms of efficient as was present there.

3.1.3. Recommendations

Teredo is NOT RECOMMENDED as a solution for managed networks. Administrators of such networks may wish to filter all Teredo traffic at the boundaries of their networks. It is sufficient to filter out the Teredo connection requests to stop further Teredo traffic. The easiest mechanism for this would be to filter out incoming traffic with source port 3544 and outgoing traffic with destination port 3544.

3.2. Problems with deep packet inspection of Teredo data packets

3.2.1. Problem

There is no efficient mechanism for network-based devices to inspect the contents of Teredo data packets, the way they can for native IPv6 packets. This makes it difficult to apply the same controls as they do to native IPv6.

3.2.2. Discussion

The only well known port that Teredo traffic uses is UDP 3544 and RFC 4380 only requires that to be used for the Teredo server service port. The client and relay components can use any port they wish.

The implication of this is that network-based devices that wish to passively inspect (and perhaps selectively apply policy to) all encapsulated Teredo-based traffic must inspect all UDP packets (or at least all UDP packets not part of session that is known not to be Teredo). This is inefficient (more so that say 6to4), especially considering that a heuristic must then be applied to determine if a packet is indeed Teredo. This may be too slow to make use of in practice, especially if it means that all UDP packets must be taken off of the device's "fast path".

One heuristic that can be used on UDP packets to determine if they are Teredo-related or not is as follows:

1. The packet is not Teredo if it is not UDP over IPv4.

2. Set T to the UDP payload offset.

3. Set E to the end of the packet plus one.

4. If E-T < 40 (the length of an IPv6 base header), the packet is not Teredo.

5. If the octets starting with T are 0x0001 (an indication of authentication data), T= T+13 plus the lengths of the client identifier and the authentication value, assuming T is the start of authentication data.

6. If E-T < 40, the packet is not Teredo.

7. If the octets starting with T are 0x0000 (an indication of origin encapsulation), T= T+8.

8. If E-T < 40, the packet is not Teredo.

9. If the octets starting with T is 0x0000 or 0x0001, loop back to step 5.

10. If the most significant nibble of the octet at T is not 6, the packet is not Teredo.

11. Assuming T is the start of an IPv6 header, set L to value of the payload length field, S to the start of the source address, and D to the start of the destination address.

12. If E-T != L+40, the packet is not Teredo.

13. If neither S nor D start with 0x20010000 (the Teredo prefix), the packet is not Teredo.

14. The packet is assumed to be Teredo, with the IPv6 header starting at T.

This is similar to the packet reception checks in [RFC4380]. The loop is present due to the possibility that some Teredo component will accept a Teredo packet even if the authentication and origin encapsulation are reversed or repeated and that either an attacker or an evasive user will use that to evade inspection. It is possible that non-Teredo packets will match as Teredo using this heuristic (in which case additional heuristics can be added), but Teredo packets should not escape inspection, absent implementation bugs.

It is not possible to monitor Teredo setup on specific ports to know to expect that Teredo traffic will appear on certain ports later since in some cases there are no Teredo setup packets (e.g., when a Teredo client is sending a packet to another Teredo client that is not behind a restricted NAT).

3.2.3. Recommendations

As illustrated above, it is very clear that inspecting the contents of Teredo data packets is highly complex and impractical. For this reason, if a network wishes to monitor IPv6 traffic, Teredo is NOT RECOMMENDED as a transition solution. As an alternative, the network may provide native IPv6 connectivity or a managed network solution like ISATAP [RFC4214]

4. Increased Exposure Due to Teredo

4.1. Teredo NAT Holes Increase Attack Surface

4.1.1. Problem

The opening created in a NAT due to a Teredo client increases its Internet attack surface area. If vulnerabilities are present, this increased exposure can be used by attackers and their programs.

4.1.2. Discussion

When a Teredo client is active, a mapped port is maintained on the NAT through which Internet hosts can send packets and perhaps establish connections. The following sequence is intended to sketch out the processing on the Teredo client host that can be reached through this; the actual processing for a given host may be somewhat different.

1. IPv4 host firewall processing

2. IPv4 processing by stack

3. UDP processing by stack

4. Teredo client processing

5. IPv6 host firewall processing

6. IPv6 processing by stack

7. various upper layer processing may follow

The firewall (and other security) processing may or may not be present, but if it is, some of the IPv6 processing may be filtered. (By the virtue of the Teredo client being active, we can infer that the IPv4 firewall is unlikely to do any filtering for this.) Any of this processing may expose vulnerabilities an attacker can exploit;

similarly these may expose information to an attacker. Thus, even if firewall filtering is in place (as is prudent) and filters all incoming packets, the exposed area is non-trivial.

The exposed area is even larger than if a native IPv6 Internet connection was in place, due to the processing that takes place before IPv6 is reached. It is also larger than for a native IPv4 connection due to the UDP, Teredo, and IPv6 processing.

One possibility is that a layer 3 targeted worm makes use of a vulnerability in the exposed processing. While the main benefit to worms from Teredo is targeting at layer 3 reaching the end host, even a throughly firewalled host could be subject to a worm that spreads with a single UDP packet if the right remote code vulnerability is present; such worms can spread quickly as evidenced by Slammer.

4.1.3. Recommendations

This problem seems inherent in Teredo being active on a host, so the solution seems to be to minimize Teredo use.

For example, it can be active only when it is really needed and only for as long as needed. So, the Teredo interface can be initially not configured and only used when it is entirely the last resort. The interface should then be deactivated again as soon as possible. Note however that the hole will remain in the NAT for some amount of time after this, so some processing of incoming packets is inevitable (unless the client's IPv4 address is changed).

4.2. Unusually High Exposure of a NAT Hole

4.2.1. Problem

Attackers are more likely to know about a Teredo client's NAT hole than a typical hole in the NAT. If they know about the hole, they could try to use it.

4.2.2. Discussion

There are at least three reasons whey an attacker is more likely to learn of the Teredo client's exposed port than a typical NAT exposed port:

1. The NAT mapping is typically held open longer and kept more stable than would otherwise be the case. This increases the chance of it being discovered.

2. The external IP address and port is contained in the client's Teredo address. While the Teredo protocol itself only distributes this address on packets, peers and even network components such as Teredo relays may record the Teredo address in, for example, log files; the address may even make its way onto, for example, peer-to-peer host advertisements.

3. The Teredo protocol contains more messages that are exchanged and with more parties than is typical, offering more chance for visibility into the port and address in use. All Teredo protocol packets contain the client's external address and port.

4.2.3. Recommendations

The recommendations from Section 4.1 seem to apply here as well: minimize Teredo use.

4.3. Teredo Bubble Facility Widens Hole in Restricted NAT

4.3.1. Problem

The bubble facility offered by clients and their servers to relays essentially turns a restricted NAT into an unrestricted one, for all Teredo client service ports. This eliminates NAT filtering for such ports and may eliminate the need for an attacker to spoof an address.

4.3.2. Discussion

Restricted NATs and port restricted NATs [RFC3489] limit the source of incoming packets to just those that are a previous destination. This poses a problem for Teredo, so [RFC4380] provides a facility for relays, upon request, to become a previous destination. This works by a "bubble" packet sent to the server, passed to the client, and then sent by the client (through the NAT) to the originator. However, any host on the Internet can use this facility, not just relays, since any host can serve as a host-only relay.

This removes any NAT-based barrier to attackers sending packets in through the client's service port. In particular, an attacker would no longer need to either be an actual previous destination or to forge its addresses as a previous destination. When forging, the attacker would have had to learn of a previous destination and then would face more challenges in seeing any returned traffic.

There may be equivalent functionality in other protocols to provide this service.

4.3.3. Recommendations

This facility is necessary for Teredo to operate, at least in its current form. Minimizing Teredo use (see Section 4.1.3) would lower the attacker opportunity related to this exposure.

5. Teredo Address Concerns

5.1. Feasibility of Guessing Teredo Addresses

5.1.1. Problem

It may be feasible guess Teredo addresses, either when looking for a specific Teredo client or when looking for an arbitrary Teredo client. This is in contrast to native IPv6 address in general.

5.1.2. Discussion

Teredo addresses are structured and some of the fields contained them are fairly predictable. This can be used to better predict the address.

Teredo prefix: This field is 32 bits and has a single IANA assigned value

Server: This field is 32 bits and is set to the server in use. The server to use is usually statically configured on the client; this may resolve to one or a small number of IPv4 addresses for the server. Certain static configurations can be reasonably expected to be common (e.g., those that are the default with a Teredo client implementation). This suggests that overall entropy of the server field will be low, i.e., that the server will not be hard to predict. Attackers could confine their guessing to the most popular server IP addresses.

Flags: The flags field is 16 bits in length, but RFC 4380 provides for only one of these bits (the cone bit) to vary.

Client port: This 16 bit field corresponds to the external port number assigned to the client's Teredo service port. Thus the value of this field depends on two factors (the chosen Teredo service port and the NAT port assignment behavior) and therefore it is harder to predict the entropy this field will have. If clients tend to use a predictable port number and NATs are often port-preserving ([RFC4787]), then the port number can be rather predictable.

Client IPv4 address: This 32 bit field corresponds to the external IPv4 address the NAT has assigned for the client port. In principle, this can be any address in the assigned part of the IPv4 unicast address space. However, if an attacker is looking for the address of a specific Teredo client, they will have to have the external IPv4 address pretty well narrowed down. Certain IPv4 address ranges could also become well known for having a higher concentration of Teredo clients, making it easier to find an arbitrary Teredo client. These addresses could correspond to large organizations that allows Teredo such as a university or enterprise or to Internet Service Providers that only provide their customers with RFC 1918 addresses.

Optimizations in scanning can also reduce the number addresses that need to be checked. For example, for addresses behind a cone NAT, it would likely be easy to probe if a specific port number is open on a IPv4 address, prior to trying to form a Teredo address for that address and port.

Most of this is elaborated on more in [TTPTPNSOSI].

5.1.3. Recommendations

The Microsoft web site [MSTO] indicates that Windows Vista and Longhorn make additional use of the flags field, beyond what the RFC specifies. 12 bits (the "A" bits in "CRAAAAUG AAAAAAAA") are chosen at random by the client at the end of qualification. Assuming there is no bias in those bit settings, then this adds 12 additional bits of entropy (4096 times as many addresses). We recommend this be formally added to the next version of the Teredo specification.

The other thing we can recommend is that the client chose the Teredo service port in as random manner as feasible, in case the NAT port assignment behavior is based on the internal port number.

5.2. Profiling Targets Based on Teredo Address

5.2.1. Problem

An attacker encountering a Teredo address has the opportunity to infer certain relevant pieces of information that can be used to profile the host before sending any packets. This can reduce the attacker's footprint and increase the attacker's efficiency.

5.2.2. Discussion

The Teredo address reveals some information about the nature of the client. The information is reasonably reliable, even if some of it

is not tied to the Teredo protocol specification.

- That a host has a Teredo address at all means that there is a Teredo client implementation available for that platform. It probably also means that it was installed by default and also that the hosts default rules for using it made it susceptible to being in use. For example, as of this writing, seeing a Teredo address strongly suggests that the host it is on is running Windows Vista.

- The server field in the Teredo address also suggests some information. Teredo client software most often get to the end user, installed, and configured using some degree of automation. It seems likely that the majority of the time the Teredo server that results from the initial configuration will go unchanged from the initial setting. Moreover, the server that is configured for use may be associated with particular means of installation, which often suggests the platform. For example, if the server field in the Teredo address is one of the IPv4 addressees that teredo.ipv6.microsoft.com resolves to, that suggests that the host is running Windows.

- The external IPv4 address in a Teredo address can of course be readily associated with a particular organization or at least an ISP.

- It is also possible that external client port numbers may be more often associated with particular client software or the operating system it is running on. The usefulness of this is reduced by the different NAT port number assignment behaviors, though the net result of this composition can not be determined without study.

The platform, Teredo client software, or organization information can be used by an attacker to target attacks more carefully. For example, an attacker may decide to use an address if it corresponds to an organization they want to penetrate. (That example would not be unique to Teredo addresses, but shows that Teredo reveals the same information.) An attacker or worm might also decide to use a Teredo address only if it looks to be associated with Windows or a certain version of Windows. (This does not seem to have a strong analogue in native IPv4 or IPv6 addresses.)

The cone bit tells the attacker whether a bubble is needed to proceed a connection. It may also have some value in terms of profiling to the extent that it reveals the security posture of the network. If the cone bit is set, the attacker may decide it is fruitful to port scan the embedded external IPv4 address and others associated with the same organization, looking for open ports.

5.2.3. Recommendations

Deprecating the cone bit would prevent the a priori revelation of the security posture of the NAT and would not reduce the functionality of the Teredo protocol.

If installation programs randomized the server setting, that would reduce the extent to which they can be profiled. Similarly, administrators can chose to change the default setting to reduce the degree to which they can be profiled ahead of time.

Randomizing the Teredo client port in use would mitigate any profiling that can be done based on the external port, especially if multiple different Teredo clients did this.

6. Additional Security Concerns

6.1. Attacks Facilitated By Changing Teredo Server Setting

6.1.1. Problem

Malware or a malicious user could change a Teredo client's server setting. This would allow them to at least monitor peer IPv6 addresses and at worst pretend to represent the remote peer.

6.1.2. Discussion

[RFC4380] documents that the Teredo server must be a trusted entity. However, it may be possible for malware or a malicious user to quietly change the Teredo client's server setting and have the user be unaware their trust has been misplaced for an indefinite period of time.

A client's server is involved in the Direct IPv6 Connectivity Test and in the bubble procedure, so it has good visibility into the client's IPv6 peers. If the server were switched to one that records this information and makes it available to third parties (e.g., advertisers, competitors, spouses, etc.) then sensitive information is being disclosed, especially if the client's host prefers Teredo over native IPv4. This is not technically difficult to set up, especially given the availability of open source Teredo server implementations. Assuming the server provides good service, the user would not have reason to suspect the change.

Full interception of IPv6 traffic could also be arranged (including pharming) which would allow any number of deception or monitoring attacks including phishing. We illustrate this with an example

phishing attack scenario.

1. A phisher stands up a malicious Teredo server (or tampers with a legitimate one). This server, for the most part, provides correct service.

2. Some malware reaches a victim host by some means and switches the host's Teredo server setting to reference the above server (either by IPv4 address or by hostname).

3. A user on the victim host types their bank's URL into his/her browser.

4. The bank's hostname resolves to both IPv6 and IPv4 addresses and the IPv6 address is selected for the socket connection. (Alternately, it just resolves to IPv6.)

5. The host is behind an IPv4 NAT so no native IPv6 or ISTAP connection is possible, so the Teredo interface is used.

6. The Teredo client uses the server for help in connecting to the the bank's IPv6 address. It asks the server to pass along an IPv6 ping so it can determine what Teredo relay to use in sending packets to the bank's IPv6 address and so it knows what relay to trust packets from for the peer.

7. The malicious server recognizes the IPv6 address as belonging to a bank that it wants to phish against, so it sends an encapsulated ping reply to the client. This is made to look like a legitimate reply sent via a Teredo relay; however the relay it is supposedly returned from is actually a phishing site. This site could even be on the same host as the malicious Teredo server.

8. The rest works pretty much like any normal phishing transaction, except that the phishing host acts as local Teredo relay, since the victim host thinks it is communicating via a Teredo relay with the bank's IPv6 address.

This pharming type attack is not entirely novel, switching DNS server settings to a malicious DNS server could have similar effect.

6.1.3. Recommendations

The scope of the attack can be reduced by limiting Teredo use in general but especially in preferring native IPv4 to Teredo-tunneled IPv6; this is because it is reasonable to expect that banks and similar web sites will continue to be accessible over IPv4 for as

long as a significant fraction of their customers are still behind IPv4 NATs.

In general, anti-phishing and anti-fraud provisions should help with aspects of this, as well as software that specifically monitors for Teredo server changes.

On the host, it should require an appropriate level of privilege in order to change the Teredo server setting and we recommend that the user be prompted when the Teredo server setting has been changed. Making it easy to see the current Teredo server setting (e.g., not requiring privilege for this) should help detection of changes.

6.2. RFC 4380 Implies That Teredo Improves Security

6.2.1. Problem

The Security Considerations section of RFC 4380 states that it can argued that Teredo improves security. The above sections argue to the contrary. This misleading or inaccurate claim can be taken out of context and used to downplay Teredo security implications.

6.2.2. Discussion

The "Security Considerations" section of [RFC4380] begins with:

> "The main objective of Teredo is to provide nodes located behind a NAT with a globally routable IPv6 address. The Teredo nodes can use IP security (IPsec) services ... without the configuration restrictions still present in 'Negotiation of NAT-Traversal in the IKE' [RFC3947]. As such, we can argue that the service has a positive effect on network security. However, the security analysis must also envisage the negative effects of the Teredo services..."

We agree that Teredo improves the ability to use IPsec in traversing a NAT and the security properties that it provides are a benefit in certain cases, specifically when the alternate session directly involves NAT translation, IPsec is desired to be used, and circumstances allow IPsec to be used. In this case the nice security properties IPsec can provide have been allowed by Teredo. However, IPsec does not solve all security problems.

It is hoped that by this point the reader will agree that Teredo introduces security risk and does not improve security overall. Hence we feel the sentence that "the service has a positive effect on network security" goes to far in stating its point, even considering the following sentence which may somewhat reduce the pointedness of

the claim. Someone may not recognize the full security impact of Teredo after reading the sentence.

6.2.3. Recommendations

We recommend that no claims regarding a positive security impact from Teredo be made, unless the scope of such a claim is immediately clear. We also recommend that the security concerns identified in this document be included in an updated Teredo standard document, except to the extent that the Teredo protocol has been improved to mitigate them.

7. Security Considerations

This document identified security concerns with Teredo that were not included in RFC 4380.

8. IANA Considerations

There are no IANA considerations from this document.

9. References

9.1. Normative References

[RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[RFC2461] Narten, T., Nordmark, E., and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, December 1998.

[RFC3489] Rosenberg, J., Weinberger, J., Huitema, C., and R. Mahy, "STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)", RFC 3489, March 2003.

[RFC4214] Templin, F., Gleeson, T., Talwar, M., and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", RFC 4214, October 2005.

[RFC4380] Huitema, C., "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)", RFC 4380, February 2006.

[RFC4787] Audet, F. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, January 2007.

9.2. Informative References

[MSTO] Microsoft, "Teredo Overview", <http://www.microsoft.com/technet/prodtechnol/winxppro/maintain/teredo.mspx>.

[TTPTPNSOSI]
Hoagland, J., "The Teredo Protocol: Tunneling Past Network Security and Other Security Implications", November 2006, <http://www.symantec.com/avcenter/reference/Teredo_Security.pdf>.

[WVNASA] Hoagland, J., Conover, M., Newsham, T., and O. Whitehouse, "Windows Vista Network Surface Analysis", March 2007, <http://www.symantec.com/avcenter/reference/Vista_Network_Attack_Surface_RTM.pdf>.

Authors' Addresses

James Hoagland
Symantec Corporation
350 Ellis St.
Mountain View, CA 94043
US

Email: Jim_Hoagland@symantec.com
URI: http://symantec.com/

Suresh Krishnan
Ericsson
8400 Decarie Blvd.
Town of Mount Royal, QC
Canada

Phone: +1 514 345 7900 x42871
Email: suresh.krishnan@ericsson.com

Full Copyright Statement

Copyright (C) The IETF Trust (2007).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).