An Approach To Secure Teredo Tunneling Technology

P. Shanmugaraja
Associate Professor, Department of IT, Sona College of Technology,
Salem, Tamil Nadu, India

D. Balamurugan
Associate Professor, Department of CSE, Sona College of Technology,
Salem, Tamil Nadu, India

S. Chandrasekar
Principal, Gnanamani College of Technology
Namakkal, Tamil Nadu, India

Abstract

The Internet Engineering Task Force (IETF) is the organization responsible for defining the Internet Protocol standards. At the time of IPv4 development by IETF, the vast Internet usage and Internet security issues were not considered. The Internet Protocol Version 6 is developed to provide new services and to support the Internet's growth. The major drawback of IPv6 is it is not backward compatible with IPv4. So, to make a communication between IPv4 node and IPv6 node a transition technology is needed. Tunneling technology is a good option to make the communication link between IPv6 and IPv4 withoutmaking any major changes in the existing infrastructure. Transition mechanism is vulnerable to security threats very easily. Teredo is one such Tunneling mechanism which achieves the communication between IPv6 and IPv4. This paper analyzes security threats that affect Teredo and solutions to overcome those security threats.

1. Introduction

THE current Internet and all the network users predominately use Internet Protocol version 4(IPv4). IPv4 address size is 32 bits. There can be only 2^{32} addresses which can address only 4 billion unique machines. Because of the rapid growth of Internet users IPv4 address are exhausted. While developing IPv4 the developers never thought about the massive growth of Internet users. IETF developed IPv4 in the year . IPv6 had been proposed at IETF as the next generation of IP at early in the 1990's. The problem with IPv6 is it's backward in compatibility with IPv4. Machines configured with IPv4 and IPv6 cannot communicate directly. But they in turn need a transition mechanism to communicate. Transition from IPv4 to IPv6 is a long term process. This transition not only affects Internet protocol but also affects other protocols operating in the network layer such as ICMP (Internet Control Message Protocol), DNS (Domain Name System), BGP (Border Gateway Protocol), OSPF (Open Shortest Path First) and RIP (Routing Information Protocol). They need to be modified or upgraded.

2. Methods in IPv6 to IPv4 translation

Interoperability between IPv4 and IPv6 is accomplished by integrating both the protocols using various transition techniques. The existing transition techniques are

1. Dual Stack
2. Tunneling
3. Translation

2.1 Dual Stack Systems

It provides support for both the network layer protocols IPv4 and IPv6. Both protocols suites work independently in this system. All the hardware and software components of this network system should support both IPv4 and IPv6 protocols. It requires current infrastructure to be compatible with IPv6 however, if the current network is not ready then it must be upgrade. It is important to understand that having a device being able to communicate over both IPv4 or IPv6 does not necessarily means that all applications operating within this device are capable of utilizing both IPv4 and IPv6. Fig 1.0 shows the dual layer architecture of a system.

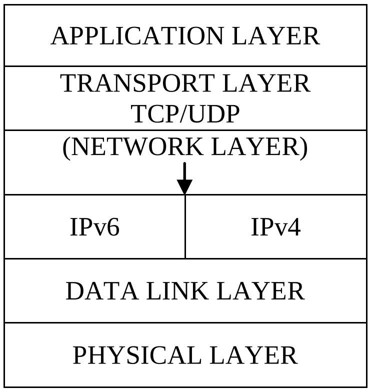

Fig 1.0 Dual layer architecture

2.2 Tunneling

The term “tunneling” refers to a means of encapsulating one version of IP in another so that the packets travel over a backbone network that does not support the encapsulated IP version. The tunneling protocol carries the tunneled protocol. Tunneling can be either IPv6-over-IPv4 or IPv4 –over-IPv6 networks. Using this technique an IPv4 user can communicate with IPv6 network using the existing IPv4 network. Tunnels are either configured tunnels or automatic tunnels. Configured tunnels require manual administration. Fig 2.0 shows IPv6 over IPv4 tunneling. Automatic tunneling is of different types as shown below in the figure 2.1.

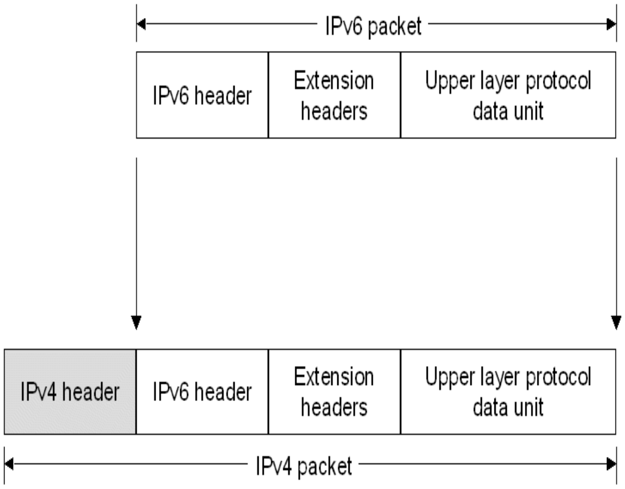

Fig 2.0 IPv6 over IPv4 Tunneling

AUTOMATIC TUNNELING	6over4
	6to4
	ISATA
	Teredo
	DSTM
	Tunnel brokers

Fig 2.1 Different types of automatic tunneling

2.3 Translation

Translator is a device capable of translating traffic from IPv4 to IPv6 or vice and versa. This mechanism intends to eliminate the need for dual-stack network operation by translating traffic from IPv4-only devices to operate within an IPv6 infrastructure. It performs Header and Address Translation between the two protocols. The advantage of this technique is IPv4 users can use this translation technology with no or little change in the existing infrastructure to connect with IPv6 network and vice versa. Some of the feature of IPv6 are lost when translation techniques and it does not solve the problem of IPv4 address space depletion. It works at different layers as shown in fig 3.0.

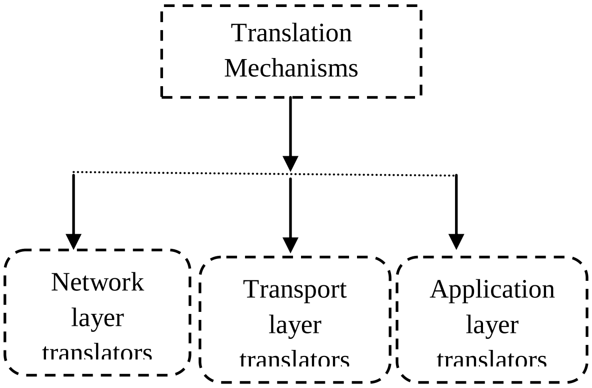
Translation Mechanisms
Network layer translators
Transport layer translators
Application layer translators

Fig 3.0 Types of Translation Mechanisms

Fig 3.0 Types of Translation mechanisms

The selection of the transition methods is site specific. There is no single best solution. Generally ISP's use Tunnel Broker, 6to4 relay, and manual tunnels. The best practice is to deploy Dual Stack systems.

3. MAJOR SECURITY CONCERNS IN TRANSITION MECHANISMS

A transition mechanism welcomes various security attacks. This section examines the attacks possible with different mechanisms. Tunneling can be used to avoid security measures. In tunneling IPv6 data is encapsulated inside an IPv4 packet before passing the traffic to the destination. Tunnel sniffing and eavesdropping is possible in the network which uses tunneling protocol. It is vulnerable to Denial-of-Service (DoS) attacks, Reflection Denial-Of-Service (DoS) and service theft, in which a malicious node or site or malicious user may make unauthorized use of the service. There is no predefining configuration between tunneling end points which leads to the above said attacks. In automatic tunneling all receiving nodes must allow decapsulation of packets that can be sourced from anywhere. The problem becomes more serious when IPv6 tunneled over IPv4 encapsulation in UDP, as UDP usually allowed passing through NAT's and Firewalls [1]. Pay load cannot be inspected in Tunnels encapsulating IPv6 in SSL/TLS or IPSec. 6to4 is the most widely used tunneling mechanism in the world for transition between IPv4 and IPv6. 6to4 routers cannot identify whether relays are legitimate [2]. 6to4 is vulnerable to packet laundering. It is subject to administrative abuse, e.g., service theft [2]. Some Operating System (Windows Vista or Windows 7) enables tunneling by default.

The following threats affect Translation technology. Circumventing ingress filtering, improper use and buffer overflow attack[8]. IPSec cannot be used in Transport Relay Translator because Translation works in the Transport layer and IPSec in Network layer. The translation system intersects TCP connection between sending and receiving hosts. This is an illegitimate behavior for a communicating node. The Transport Relay Translator must retain state, so it is vulnerable to various DOS attacks [3]. In TRT Protocols based on IP authentication does not work.

Duplication of processes is unavoidable in Dual stack technologies. It has to face the threats of both IPv4 and IPv6. It relies on tunneling and translation mechanisms for interoperability of networks that are not dual stack [4]. Unexpected tunneling between the hosts may occur which may violate security policies.

4. Threats Due to Transition Mechanisms

There are a large number of transition mechanisms to deploy IPv6, but can broadly be categorized into, dual stack, tunneling (manual/automatic), and translation [1]. A dual-stack node has complete support for both IPv6 and IPv4. The two protocol stacks work independently but coexist within the same network, so applications can be subject from attack from both IPv4 and IPv6. Therefore, any security controls, such as firewalls, IDSs, VPN clients and so on, must be mirrored in both protocol deployments to provide full protection. Tunneling involves the transportation of IPv6 packets over the existing IPv4 infrastructure. This usually involves encapsulating IPv6 packets within the payload of an IPv4 packet. Such mechanisms as 6to4 tunneling [4] adopted this procedure, where the security considerations have been well documented in RFC 3964 [4]. Tunneling can be used to evade security devices, so administrators must ensure they have both an IPv6 and IPv4 firewall where their rules are mirrored for both protocols. The problem is amplified when IPv6 is tunneled over IPv4 encapsulated in UDP, as UDP is usually allowed to pass through NATs and firewalls [4]. Consequently, allowing an attacker to punch holes within the security infrastructure. The author recommends that if the necessary security measures cannot be taken, tunnelled traffic should be used with caution if not completely blocked. To provide ingress and egress filtering of known IPv6 tunneled traffic, perimeter firewalls should block all inbound or outbound IPv4 Protocol 41 traffic. For circumstances where Protocol 41 is not blocked it can easily be detected and monitored by the open-source IPv4 IDS Snort.

5. **Teredo Tunneling**

Teredo is a transition technology that provides IPv6 connectivity for IPv6-enabled systems which are connected to the Internet using IPv4 protocol but which have no direct connection to an IPv6 network. It is able to provide connection even behind network address translation (NAT) devices [7]. But this cannot be possible with other transition technologies. Teredo encapsulates IPv6 datagram packets within IPv4 UDP packets to make communication. It's a platform independent tunneling protocol. Teredo is designed as a final transition technology and it will exist throughout the migration of IPv4

6. Node types

Teredo defines several different kinds of nodes [7].

6.1 Teredo client

A host which has IPv4 connectivity to the Internet from behind a NAT and uses the Teredo tunneling protocol to access the IPv6 Internet. Teredo clients are assigned an IPv6 address that starts with the Teredo prefix (2001:0::/32).

6.2 Teredo server

A well-known host which is used for initial configuration of a Teredo tunnel. A Teredo server never forwards any traffic for the client (apart from IPv6 pings), and has therefore very modest bandwidth requirements (a few hundred bits per second per client at most), which allows a single server to support large numbers of clients. Additionally, a Teredo server can be implemented in a fully stateless manner, thus using the same amount of memory regardless of how many clients it supports.

6.3 Teredo relay

The remote end of a Teredo tunnel. A Teredo relay must forward all of the data on behalf of the Teredo clients it serves, with the exception of direct Teredo client to Teredo client exchanges. Therefore, a relay requires a lot of bandwidth and can only support a limited number of simultaneous clients. Each Teredo relay serves a range of IPv6 hosts (e.g. a single campus/company, an ISP or a whole operator network, or even the whole IPv6 Internet); it forwards traffic between any Teredo clients and any host within said range.

6.4 Teredo host-specific relay

A Teredo relay whose range of service is limited to the very host it runs on. As such, it has no particular bandwidth or routing requirements. A computer with a host-specific relay will use Teredo to communicate with Teredo clients, but it will stick to its main IPv6 connectivity provider to reach the rest of the IPv6 Internet.

7 Limitations

Teredo is not compatible with all NAT devices. Using the terminology of RFC 3489, full cone, restricted and port-restricted NAT devices are supported. It doesn't support symmetric NATs. Because of less security features Teredo support for symmetric NAT was eliminated. Teredo uses the same mapped external UDP numbers when two clients exchange encapsulated IPv6 packets. With this technology it can establish a direct communication between clients. A relay has to be used to achieve triangle routing. A Teredo implementation tries to detect the type of NAT at startup, and will refuse to operate if the NAT appears to be symmetric. This limitation can sometimes be worked around by manually configuring a port forwarding rule on the NAT box, which requires administrative access to the device. Teredo can only provide a single IPv6 address per tunnel endpoint [7]. A single Teredo tunnel cannot be used to connect multiple hosts.

8 Security issues in Teredo Tunneling

8.1 Teredo Client to NAT

This attack manipulated a Teredo tunnel. NAT and a forwarding node (a router, a firewall, a Mobile IP home agent etc.) that uses Teredo are the victims of this attack. The NAT is of type cone and it supports hair-pin routing with source address translation. These two assumptions are based on two requirements, REQ-8 and REQ-9, included in a Best Current Practice published by the IETF [5]. It is initiated by sending an IPv6 packet over the Teredo tunnel. The packet's destination IPv4 address and UDP port are the same as the source IPv4 address and UDP port[8]. They are equal to the external IPv4 address and UDP port of the client. The IPv6 destination and source addresses are Teredo addresses, denoted by IPd Teredo and IPs Teredo, respectively, where the fields <obfuscated external port>and <obfuscated external IP> in both addresses are identical and equal to the 1's complement of the Teredo client's external port and address, respectively[6]. The fields <Teredo server> or <flags> in those addresses should be different. Although their values are not important, they must not be equal to the respective fields in the client's Teredo address. Consequently, IPd Teredo and IPs Teredo are not equal to the client's Teredo address.

Having a state associated with the client following the initial qualification procedure and being of type cone, the NAT will not filter the attack packet and will pass it to the internal network while translating the destination IPv4 address and UDP port to the internal address and port of the client. The packet reaches the client over its IPv4 interface. The IPv4 source address and port of the packet correspond to the IPv6 source Teredo address; hence the client will admit the packet and remove the IPv4 and UDP headers. Since IPd Teredo is not the address of the client and the client is in forwarding mode, the client forwards the packet back to the network through its Teredo interface. The packet is encapsulated again with IPv4 and UDP headers, while the destination address and port are derived from IPd Teredo. Namely, they are equal to the client's external address and port. The source address and port are the client's internal address and port. Since the NAT is assumed to support hair-pin routing, when the packet reaches the NAT it will be routed back to the internal network. The destination 5 address and port will be translated to the client's internal address and port. Since the NAT supports source address translation, the source address and port will be translated to the client's external address and port. The resulting packet is identical to the previous packet. Hence, it will be routed back to the client, in which the loop will start again. In this attack the Hop Limit field will decrease only when the packet traverses the Teredo client. Only then is the packet handled by an IPv6 stack. In all the other hops on the loop, including the NAT, only IPv4 processing takes place. Initial attack packet can be prevented by using ingress filtering methods [6].

8.2 Teredo Server

This attack differs from the attacks above. First, it engages with only one victim, a Teredo server. Second, the loop is not formed by forwarding the same IPv6 packet over and over, but by creating a new packet over and over again. Hence, the lifetime of the loop is infinite and not limited by the Hop Limit field. These two differences make this attack the most violent of all the attacks described in this paper. Executing the attack on a victim will result in an immediate exhaustion of the victim's CPU resources and will bring it to a crawl. The attack loop is formed by tricking a Teredo server to produce a bubble destined to itself upon receipt of another bubble. The attack is depicted in. It is initiated by sending a bubble over the Teredo tunnel to the server[9]. The bubble's destination IPv4 address and port are identical to its source IPv4 address and port. They are equal to the IPv4 address of the server and 3544, respectively. The IPv6 destination and source addresses are two distinct Teredo addresses, in both of which the fields <obfuscated external port> and <obfuscated external IP> are identical and equal to the 1's complement of the server's IP and port (3544). The server receives and processes the packet as a normal Teredo bubble. In particular, it verifies that the source IPv4 address and port correspond to the source IPv6 Teredo address. The server then creates a new bubble destined to the IPv4 address and port as derived from the IPv6 destination address. The Teredo specification does not define a check to prevent this [10]. Hence, the bubble will be destined to the server's IPv4 address and to port 3544. Since the new bubble is identical to the previous one, the loop starts again indefinitely [11].

9 Mitigating measures

This is a simple solution to prevent the threat to some extent. If the following rule is applied to the Teredo then it can be safeguarded. In Teredo tunneling if the receivers address is a Teredo address, then the field <obfuscated external IP> must not be equal to the 1's complement of an IPv4 address of one of the node's interfaces or to an IPv4 address which is mapped to that node by a NAT2.

Conclusion

IPv6 protocol will replace IPv4 protocol completely in the coming years. Till then we need translators to make communication between IPv6 and IPv4. Even though IPv6 solves most of the problems of IPv4, it also introduces new network security issues. In this paper presents an overview of transition mechanisms and their security threats. It mainly focuses on security issues of IPv6 automatic tunneling mechanism Teredo. The proposed mitigation measures for threats are relatively simple and more research should be carried out to find a complete solution. The current Internet is growing very vast and difficult to manage which leads to security threats every second.

References

[1] S. Convery and D. Miller, "IPv6 and IPv4 Threat Comparison and Best-Practice Evaluation(v1.0)," Mar. 2004; www.seanconvery.com/v6-v4-threats.pdf.

[2] P. Savola,C. Patel, "Security considerations for 6to4", IETF RFC 3964, DEC 2004

[3] J. Hagino, K. Yamamoto, “An IPv6- to- IPv4 Transport Relay Translator”, IETF RFC 3142, JUN 2001

[4]E. Nordmark, R. Gilligan Intransa, “Basic transition mechanisms for IPv6 hosts and routers “ ,IETF RFC 4213, OCT 2005

[5] F. Audet *et al.*, “Network address translation (NAT) behavioral requirements for unicast UDP,” IETF RFC 4787 (BCP 127), January 2007.

[6] F. Baker and P. Savola, “Ingress filtering for multihomed networks,” IETF RFC 3704, March 2004.

[7] C. Huitema, “Teredo: Tunneling IPv6 over UDP through network address translations (NATs),” IETF RFC 4380, February 2006.

[8] P. Shanmugaraja, S. Chandrasekar, “Accessible methods to mitigate security attacks on IPv4 to IPv6 transition”, European Journal of Scientific research, March 2012 pp 165-173.

[9] S. Kent and R. Atkinson, “Security architecture for the internet protocol,” RFC 2401 (Proposed Standard), Internet Engineering Task Force, Nov.1998, updated by RFC 3168.

[10] S. Kent and R. Atkinson, “IP encapsulating security payload,” RFC 2406(Proposed Standard), Internet Engineering Task Force, Nov. 1998.

[11] S. Kent and R. Atkinson, “IP authentication header,” RFC 2402 (Proposed Standard), Internet Engineering Task Force, Nov. 1998.

[12] D. Harkins and D. Carrel, “The Internet Key Exchange (IKE),” RFC 2409 (Proposed Standard), Internet Engineering Task Force, Nov. 1998.