Chapter 9

Security

This chapter provides an overview of the current security issues in an IPv6 based networking environment and suggests a number of helpful security “guidelines”.

First we will analyse how the IPv6 changed the security of IP networking environment. We will concentrate on the threat analysis compared with IPv4. Then we will discuss the major building block of a security architecture: IPv6 firewalls. Finally we will discuss the security implications of deploying various IPv4-IPv6 co-existnce and transitioning mechanisms.

9.1 What has been Changed in IPv6 Regarding Security?

In this section will enumerate the different threats that you can face when you operate a IP networking environment and we trying to provide some sort of solution in IPv6 in mind.

9.1.1 IPSec

IPSec is a framework of open standards developed by the Internet Engineering Task Force (IETF) that provide security for transmission of sensitive information over unprotected networks such as the Internet. IPSec acts at the network layer, protecting and authenticating IP packets between participating IPSec devices (peers), such as Cisco routers. IPSec provides the following network security services. These services are optional. In general, local security policy will dictate the use of one or more of these services:

- Data confidentiality—The IPSec sender can encrypt packets before sending them across a network.
- Data integrity—The IPSec receiver can authenticate packets sent by the IPSec sender to ensure that the data has not been altered during transmission.
- Data origin authentication—The IPSec receiver can authenticate the source of the IPSec packets sent. This service is dependent upon the data integrity service.
- Anti-replay—The IPSec receiver can detect and reject replayed packets.

With IPSec, data can be sent across a public network without observation, modification or spoofing.

IPSec functionality is essentially identical in both IPv6 and IPv4; however, IPSec in IPv6 can be deployed from end-to-end - data may be encrypted along the entire path between a source node and destination node. (Typically, IPSec in IPv4 is deployed between border routers of separate networks.) In IPv6, IPSec is implemented using the authentication extension header and the ESP extension header. The authentication header provides integrity and authentication of the source. It also provides optional protection against replayed packets. The authentication header protects the integrity of most of the IP header fields and authenticates the source through a signature-based algorithm. The ESP header provides confidentiality, authentication of the source, connectionless integrity of the inner packet, anti-replay and limited traffic flow confidentiality.

9.1.2 IPv6 Network Information Gathering

Usually an attacker begins his/her activity by network, host and service reconnaissance, most often by scanning. This is typically done via some sophisticated scanning methods (e.g. stealth scanning) to provide information to enable other forms of attacks. The IPv6 networking architecture provides some protection against scanning. The large number of potential hosts in a typical IPv6 LAN makes host and service identification ("fingerprinting", port scanning) quite difficult if not impossible. The exhaustive scanning of a /64 subnet is incredibly time consuming: If you have scanner that is capable of scanning 1 million addresses each second (note: the capability of today's scanners are couple of thousands address per seconds), then scanning would take 2^{64} addresses / 1000000 addresses-per-second /60 seconds-per-minute / 60 minutes-per-hour / 24 hours-per-day / 365.25 day-per-year = ~ 584,000 years!

A number of issues however could simplify the scanning process and setting important systems in danger:

Predictable addressing scheme

It is very common practice of system administrators to use specific, predictable, numbering schemes for important systems (e.g. routers, servers, etc.). The administrators should carefully select numbering pattern for their systems to help relieving with this problem.

Reducing the number of address by exploiting the structure of EUI-64 addresses

Usually the last 64 bits of the IPv6 addresses are constructed based on the modified EUI-64 algorithms as described in RFC 3513 from the IEEE 802 48 bit MAC address. In the algorithm there is padding with hexadecimal values 0xFF and 0xFE, that will reduce the problem space. The attackers can even further reduce the problem space if they guess or know in advance the vendor of the IEEE 802 network card, since the IEEE 802 addresses are constructed from a 24 bit vendor or company id and a 24 bit vendor supplied id to ensure uniqueness. (In this later case the attackers could scan the network in 17 seconds if they have a 1000000 addresses-per-second super scanner).

Scanning from inside the LAN

The possibility of information gathering on existing systems from poorly secured routers, gateways, DHCPv6 servers or other network devices. This problem is rather a system security one and the solution does not differ under IPv6: careful and timely security management, ensuring that the system is adequately protected from current threats.

Inappropriate filtering of incoming scanning messages

There is a need for particular ICMPv6 messages to be allowed in the protected network for the IPv6 protocol to operate correctly. As in IPv4, these packets can be used for information gathering therefore the security policy should be appropriately adjusted to cope with the new protocol features, allowing through only the necessary types of messages

Inappropriate filtering of multicast messages

Some IPv6 multicast addresses are used to reach group devices of the same type for convenience, e.g. all routers, all NTP servers etc. An attacker able to access these addresses could acquire access to the corresponding devices and perform attacks against them (e.g.

DoS). Careful border filtering should prevent the particular addresses from being announced or accessed outside the network's administrative borders.

Other forms of finding potential targets

The attacker also can find out potential targets by simply setting up services as honeypot to harvest addresses and after certain amount of time analyse the access logfiles of services to find out potential targets. The hosts can be identifiable this way from the log files, however if proper filtering is set up at the end-site the attacker will not get access to the potential targets.

Finally, a well known practice that is proven to be valuable under IPv4, filtering of unneeded services at the network's access points, can be equally useful under IPv6 for mitigating reconnaissance threats.

9.1.3 Unauthorised Access in IPv6 networks

Determining who has authorized access to a computer system is a policy decision. If this authorisation is enforced in TCP/IP at Layer 3 or Layer 4 then it is usually implemented in firewalls. Policy implementation in IPv6 at Layer 3 and Layer 4 is still implemented in firewalls with some design considerations.

The filtering of packets whose source (or possible destination) address should never appear in Internet routing tables (often called bogons) (e.g. non routable, non assigned etc.) is the minimal filtering that firewalls should provide. In IPv4 it is easier to filter out (deny) packets originating from bogon routes, while in IPv6 it is easier to allow legitimate packets as shown in table Table 9-1.

Table 9-1 Bogon Filtering Firewall Rules in IPv6

Rule	*Meaning*
deny 2001:db8::/32 any	Filter out documentation prefixes
allow 2001::/16 any	Allow RIR allocated prefixes 1
allow 2003::/16 any	Allow RIR allocated prefixes 2
allow 2002::/16 any	Allow 6to4 relay prefix
allow 3ffe::/16 any	Allow 6Bone prefixes - deprecated after 6th June 2006
deny any any	Deny everything else

More detailed discussion about IPv6 firewalls can be found in section 9.2 “IPv6 Firewalls”.

Of course there is also the possibility of preventing unauthorised access to the IPv6 network below the network layer. A port-based authentication mechanism such as 802.1x [8021x] is a sound way to organise a secure network infrastructure. An 802.1x based infrastructure can integrate both wired and wireless segments of an organisation's network. For more information on using 802.1x with IPv6 wired and/or wireless networks please refer to [D4.2.2].

9.1.4 Spoofing in IPv6 Networks

Most of the occurrences of various Denial of Service (DoS) attacks which have employed forged or spoofed source addresses have proven to be a troublesome issue for Internet Service Providers and the Internet community overall. RFC 2827 [RFC2827] recommends a simple, effective, and straightforward method for using ingress traffic filtering to prohibit DoS attacks which use forged IP addresses propagated from ‘behind’ an Internet Service Provider’s (ISP) aggregation point. The method, called “ingress filtering” can only prevent spoofing of the source address. An important benefit of implementing ingress filtering is that it enables the originator to be easily traced to its true source, since the attacker would have to use a valid, and legitimately reachable, source address.

The ingress filtering is usually implemented at ISP edge routers with various methods, either via firewall filters or by enforcing the uRPF (unicast reverse path forwarding) check. The behaviour of the ingress filtering is the following:

```
# uRPF processing


IF (packet's source address is from network residing behind the interface
where the packet comes from) {
    forward as appropriate
 } ELSE IF    (packet's source address is anything else) {
    deny packet
}
```

A similar technique can be implemented by the end-user of an ISP to prevent sending packets that do not belong to their network, usually called egress filtering.

These techniques can also be implemented in IPv6. IPv6 can make the ingress filtering easier, since only one prefix should be configured for the ingress filter, due to the hierarchical aggregation of IPv6 addresses. Usually only one /48 has to be configured, if you cannot setup automatically the anti-spoofing or uRPF (unicast Reverse Path Forwarding) check.

The egress filtering configuration is very similar to the ingress filtering configuration, the difference being that it is configured at the user’s equipment.

We should note, that ingress and egress filtering might be more complex, albeit not impossible if multihoming and multiple address prefixes are employed at the user site. In this case the multiple address prefixes should be appropriately configured.

9.1.5 Subverting Host Initialisation in IPv6 Networks

In IPv4 environments it is rather easy to perform attacks against the ARP protocol, since hosts cannot prove ownership of their MAC addresses. Therefore it is easy to hijack the default router on the subnet. You can protect your network on a switch by enforcing a specific number of source MAC address for all frames received on a specific port. This protection is available on some switches (notably on modern Cisco Catalysts) as a feature called port security.

If we are using DHCP for initialising hosts, the attacker on the link can perform various attacks against the DHCP server: operating a false DHCP server and delivering DHCP messages faster than the original official DHCP server, exhausting resources of DHCP server by issuing large number of requests, exhausting leased IP address space by requesting too many IP addresses etc. You can protect your system against such an attack by a combination of port security, DHCP snooping, and DHCP message rate limiting. By using port security you can prevent rogue DHCP server operation and faking different MAC addresses on a certain port. The DHCP snooping provides security by filtering untrusted DHCP messages and by building and maintaining a DHCP snooping binding table. This binding table can be used to prevent IP spoofing by only allowing IP addresses that are obtained through DHCP snooping on a particular port.

The host neighbourship in IPv6 environments also can be attacked in a similar way to ARP. Possible attacking techniques could be: sending false Neighbour Advertisement messages, performing Denial of Service against the Duplicate Address Detection procedure, or sending fake Router Advertisements as described in RFC 3756 [RFC3756]. To mitigate attacks against the Neighbour Discovery procedure you can deploy Secure Neighbour Discovery (SEND) [RFC3971]. More detailed discussion about Secure Neighbour Discovery can be found in section 9.3.

9.1.6 Broadcast Amplification in IPv6 Networks

There have been several broadcast amplification attacks against IPv4 network infrastructures. The most famous was the smurf attack where the attacker sent out the packet with following content:

Table 9-2 Structure of the Smurf Attack Packets

Spoofed address of attack target	Subnet broadcast address of amplifier network	ICMP echo

There were two problems that allowed the smurf attack to work:

1. Ingress filtering was not implemented which allowed spoofing the source address field of the attack packet.
2. The host operating systems answered to a message destined to a broadcast address.

Such a problem cannot be foreseen in IPv6 environment for various reasons:

There is no broadcast address in IPv6 environment

This would stop any type of amplification/smurf attacks that send ICMP packets to the broadcast address. However global multicast addresses for special groups of devices, e.g. link-local addresses, site-local addresses, all site-local routers, etc. are available to reach groups of devices.

The IPv6 specification does not allow answering to multicast destinations

IPv6 specifications forbid the generation of ICMPv6 packets in response to messages to global multicast addresses except in two case as described in RFC 1885 [RFC1885]:

1. The Packet Too Big Message - to allow Path MTU discovery to work for IPv6 multicast
2. The Parameter Problem Message, Code 2 - reporting an unrecognized IPv6 option that has the Option Type highest-order two its set to 10.

9.1.7 Attacks Against the IPv6 routing Infrastructure

The primary purpose of IP routing attacks are to disrupt/corrupt router peering or routing information in order to cause denial of service attack or provide help to other type of attacks like DNS cache poisoning etc.

In an IPv6 environment the operators of the network can face similar attacks against the routing infrastructure. However in an IPv6 environment the network designers should be aware of certain idiosyncrasies of IPv6 routing.

In the cases of BGP, IS-IS and EIGRP the security algorithms of the routing protocol remains the same: keyed MD5 digest. So in these cases the same protection for the routing protocol should be used.

By contrast, in the case of OSPFv3 [RFC2740] and RIPng [RFC2080] the routing protocol has been adapted to IPv6 and relies on the IPSec protocol. So if you are using OSPFv3 or RIPng for routing you should also configure IPSec to protect these routing protocols.

The other types of attacks against the routing infrastructure as mentioned are very similar to IPv4 routing infrastructure attacks, therefore similar countermeasures should be implemented: e.g. infrastructure protection, limiting access to the router, SSH authentication etc.

9.1.8 Capturing Data in Transit in IPv6 Environments

Capturing unprotected data in IPv6 networking environment very much like sniffing in IPv4 environment. Ethereal, a tool already widely used by system/network administrators in various platforms and networks is such an example. Other similar tools sporting IPv6 capabilities and not limited to “passive sniffing” of the physical layer, are soon expected to appear, if they do not exist already. The conclusion is that this type of attack presents similar to IPv4 and real threat for services over IPv6.

However, the mandatory support of IPSec in IPv6 environment might help resolving the problem since the infrastructure to protect any kind of communication (e.g. SQL database queries) is built into the systems, and can be protected against sniffing

9.1.9 Application Layer Attacks in IPv6 Environments

These days the most common attacks against computer systems are targeted at the application layer. Often these attacks gain access to system resources by exploiting buffer overflows in the applications or by gaining elevated privileges by executing code with inappropriate checking.

These types of attacks are not bound to any underlying network protocol, so we can not expect any changes if we are deploying IPv6. The operators of the services, must be aware of the problems, and update their systems to prevent such an attacks to happen.

9.1.10 Man-in-the-middle Attacks in IPv6 Environments

Without application of IPSec, any attacks utilizing Man-in-the-middle techniques will have the same likelihood in IPv6 as in IPv4.

If we keep in mind that IPSec is strongly linked to IPv6, its usage alone would be enough to avoid any problems regarding connection hijacking attempts. Unfortunately, the dominant practice we see today in terms of IPv6 deployment already in place is that network operators do not make any use of IPSec. The use of certificates can also provide the needed end-to-end authentication at the application level (e.g. Web servers). Without such end-to-end security mechanisms, a man-in-the-middle hijack is a possibility.

9.1.11 Denial of Service Attacks in IPv6 Environments

Flooding attacks are identical between IPv4 and IPv6, so preventing them in a IPv6 network will be a future challenge. This requires powerful DoS detection tools, which can analyse IPv6 communication flows to find out DoS flows.

If the communication is authenticated by IPSec, the Denial of Service packets are not delivered to the final application, but the communication channels might still be filled, which can deny legitimate users access to services.

9.2 IPv6 Firewalls

In the 1990s, firewalls became the building block of each IP network. The recent growth of IPv6 usage has necessitated analysing whether the new protocol can provide enough security without the use of IPSec. This analysis is also important since the application of IPSec on the Internet is relatively scarce and probably will be limited due to deployment difficulties of the public key infrastructure, and in spite the fact that IPSec itself provides a good, modular framework. This section tries to analyse what is available and what is missing for effective IPv6 firewalling.

The Internet firewall is a system that implements and enforces the security policy between two networks: usually protects an internal private network (Intranet) from external Internet threats. Sometime firewalls are also implemented with more than two network interfaces, where the third, fourth interfaces are used for special purposes like DMZs (DeMilitarised Zones), etc.

The firewalls usually can be operated at different levels in the networking hierarchy:

IP level	Packet filtering firewalls
Transport level	Circuit oriented firewalls
Application level	Application level proxies. There is a higher level of support in the application level proxies, e.g. transparent proxies and modularisation

The most important principle of firewalls, however, is function in helping to enforce the security policy (administrative rules) that will protect certain assets. The majority of modern firewalls employ a mix of protective methods at different levels.

In IPv6 the levels are not changed, therefore we can expect that firewalls should support IPv6 at any level. A good firewall implementation should be IP version agnostic at transport or application levels.

We will focus our discussion to packet filtering firewalls for two reasons.

1. These types of firewalls are the basic elements for the more advanced firewalls. They have become necessary components due to the very large number of existing protocols on the Internet (e.g. a wide variety of H.323 related standards, instant messaging protocols, even FTP) that prevents the operation of proxy services for every one of them
2. Currently there are only very few application level firewalling solutions available on the market that offer IPv6 capabilities.

9.2.1 Location of the Firewalls

Traditionally the firewalls are installed next to the interconnecting device (usually routers) in order to choke the unwanted traffic as close to the originating point as possible. Nowadays the firewalls (usually more then one at each network) are installed in front of the device or network, which must be protected. What are the implications of enabling IPv6 on these firewalls [Moh01], [Moh04].

- The firewalls should support Neighbour Discovery ICMPv6 message processing – This issue is rarely discussed with IPv4 firewalls: The IPv4 firewalls must support ARP protocol. The Neighbour Discovery Protocol (RFC 2461) is an extension of ARP for IPv6, therefore IPv6 firewalls must support Neighbor Discovery Protocol filtering "out of the box".
- The IPv6 firewalls should not filter out packets with proper fragmentation header. A common practice in IPv4 firewalls, to guard against the tear-drop attack or other cases of heavily

fragmented packets, is to reassemble the IP fragments at the firewalls themselves and send the complete and sanitised resulting packets to the end systems. Unfortunately this is not possible in IPv6, since fragmentation and reassembly can happen only on the originating and destination node. However, some protection which might be possible in IPv6 is discussed later.

- IPv6 firewalls must support extension headers.

The rest of the requirements are depending on the location of the firewall boxes and routers.

9.2.1.1 Internet-router-firewall-protected network architecture

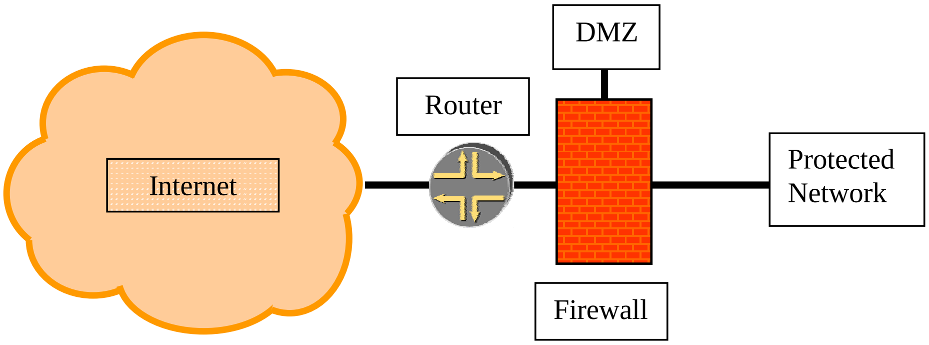

Figure 9-1 Internet-router-firewall-protected Network Setup

Additional requirements:

- In this setup the Firewall must support Stateless Address Autoconfiguration mechanisms (RFC 2462) if the Autoconfiguration option is used on the Protected Network. If the Firewall is operated transparently to the IP layer, then it should allow the Router Solicitation messages coming from hosts and their respective answer coming from the Router. It should also allow periodic Router Advertisement messages to go from the Router to the Protected Network. If the Firewall is operated non-transparently to the IP Layer, then it should be able to answer Router Solicitation messages and periodically announce Router Advertisement messages. These settings are also important if the network is operated with DHCPv6 (or other Statefull Address Assignment methods), since the Stateless RA messages will inform nodes on the network about the configuration method that is to be followed.

- If IPv6 multicast is implemented in the Protected Network, then the Firewall must support the Multicast Listener Discovery Protocol in order to keep track of the interested nodes in the Protected Network for a particular multicast group.

9.2.1.2 Internet-firewall-router-protected network architecture

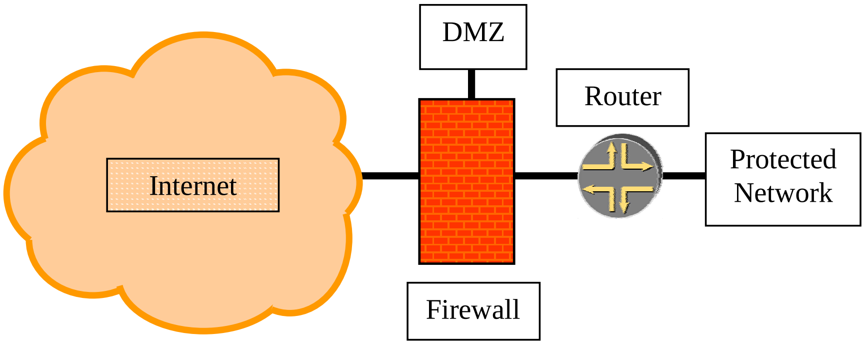

Figure 9-2 Internet-firewall-router-protected Network Setup

Additional requirements:

- The Firewall must support the dynamic routing protocol filtering, that is used by the access router (Router) and Internet Service Provider (e.g. OSPFv3, IS-IS, RIPng, or BGP). This might be challenging if IPSec is used for securing the routing protocols. As a general rule we recommend to use either static routing or BGP for such a setup, since BGP is using MD5 hash and TTL hack for securing routing updates that are IP version agnostic.

This setup might be inconvenient, since the Firewall should support a number of different access technologies, therefore it may need to support a wide variety of interfaces. This problem is expected to be less common in the future since many providers prefer handing over the Internet service over Ethernet media.

9.2.1.3 Internet-firewall/router (edge device)-protected network architecture

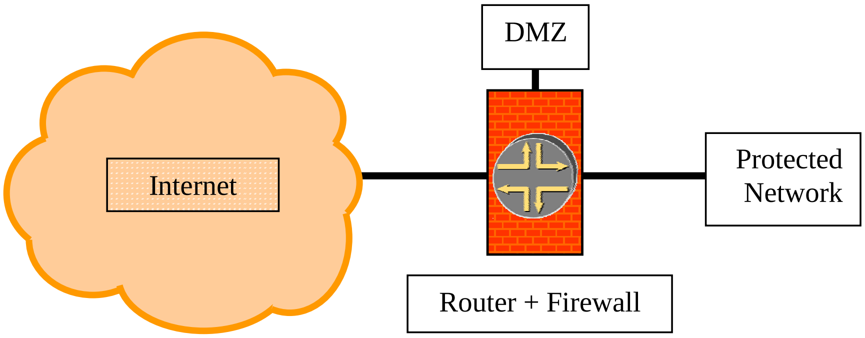

Figure 9-3 Internet-edge-protected Network Setup

Additional requirements:

- Must both support what is necessary for the previous two architecture (Router Solicitation, Router Announcement, and Dynamic routing filtering)

This is a rather powerful architecture, since it allows concentrating both the routing and the security policy in one device; however this concentration makes the particular architecture less susceptible to the security problems:

- More functionality should be integrated into one device. That makes it more complex and opens the possibility of more security problems
- Since it is only one device the principle of security: protect your network/service with more than one asset, cannot be fulfilled.

This setup is very common in home or small office environments, where a single xDSL, or cable router provides connectivity and in the same time enforces the security policy defined by the network administrator.

9.2.2 ICMP Filtering

It is very common (although questionable) practice to filter completely the ICMP messages in IPv4. This is no longer possible with IPv6. As the name that it stands for suggests, Internet Control Message Protocol for IPv6 (RFC 2463) is the control and foundation protocol for the operation of IPv6, not an auxiliary protocol that can be easily omitted. Our recommendation is the following:

ICMPv6 echo request and reply (Types 128 and 129):

- You should consider enabling at least outgoing ICMPv6 echo request and their answers, the ICMPv6 echo reply packets to facilitate debugging. Of course, it is wise to rate limit ICMPv6 debugging packets to a certain level.
- You may consider enable incoming ICMPv6 echo request packets and their answers to your well know IPv6 service machines. You should be sure, however that your IPv6 service machine can handle ICMPv6 requests over a certain rate. Of course, it is wise to rate limit ICMPv6 debugging packets to a certain level.

ICMPv6 destination unreachable (Type 1):

- You should consider enabling incoming ICMPv6 destination unreachable messages as answers, to outgoing IPv6 packets that have been sent for debugging purposes.
- You may generate proper ICMPv6 destination unreachable messages for all filtered packets. This is useful for debugging. It is a common practice in IPv4, to refrain from generating ICMPv6 destination unreachable messages to hide the networking/service structure. You can apply the same rule to IPv6. If you generate ICMPv6 destination unreachable messages, however, do it properly, setting the right reason code: no route to destination, administratively prohibited, beyond scope of source address, address unreachable, port unreachable.

ICMPv6 packet too big (Type 2):

- You must enable incoming ICMPv6 packet too big messages as answers to outgoing IPv6 packets for the Path MTU discovery to operate properly.
- You must generate ICMPv6 packet too big messages properly if your MTU is different anywhere within your network from the MTU on the link between you and your provider. So be prepared, to forward ICMPv6 packet too big messages at the firewall.

ICMPv6 time exceeded (Type 3)

- You must/should enable incoming ICMPv6 time exceeded messages to be able discover destination systems not reachable due to a low TTL value in the outgoing packets.
- You must generate correct ICMPv6 time exceeded messages since they are essential for proper operation of Internet.

ICMPv6 parameter problem (Type 4):

- You should consider enabling incoming ICMPv6 parameter problem messages as answers to outgoing IPv6 packets for debugging purpose.
- You must generate correct ICMPv6 parameter problem messages since they are essential for proper operation of Internet.

ICMPv6 Neighbour Solicitation and Neighbour Advertisement (Type 135 and 136):

- You must enable incoming and outgoing ICMPv6 Neighbour Solicitation, Neighbour Advertisement packets, with proper link-local addresses or multicast addresses for the Neighbour Discovery function to operate properly.

ICMPv6 Router Solicitation and Router Advertisement (Type 133 and 134):

- If the Stateless Address Autoconfiguration function is used, you must enable outgoing ICMPv6 Router Advertisement packets, with proper link-local addresses and multicast addresses (All node multicast addresses should be ff02::1).
- If the Stateless Address Autoconfiguration function is used, you must enable incoming ICMPv6 Router Solicitation packets, with proper link-local addresses and multicast addresses (All router multicast addresses should be ff02::2).

ICMPv6 redirect (Type 137)

- You may disallow ICMPv6 router redirect messages passing, if you have only one exit router. However, router redundancy might be implemented by router redirect. It is important to know that redirect has link-local meaning only.

ICMPv6 MLD listener query, listener report and listener done (Type 130, 131 and 132):

- You should enable incoming and outgoing ICMPv6 MLD messages, with proper link-local addresses or multicast addresses if you want to use IPv6 multicast on a bigger scope than link-local. This is required if the "internet-router-firewall-protected network" architecture is used. In this case your firewall should act as an MLD router.

ICMPv6 renumbering (Type 138)

- You may disallow ICMPv6 router renumbering messages passing, since router renumbering is not widely adopted.

ICMPv6 node information query and reply (Type 139 and 140)

- You may disallow ICMPv6 node information query and reply processing, since node information query/reply is not widely adopted.

We summarise the ICMPv6 recommendations in Table 9-3.

Table 9-3 ICMPv6 Recommendations

ICMPv6	Usage
Echo request/reply	Debugging
Destination unreachable	Debugging – better indicators
TTL Exceeded	Error report
Parameter problem	Error report
NS / NA	Important for IPv6 Neighbour Discovery.
RS / RA	For Stateless Address Autoconfiguration
Packet too big	Important for PATH MTU discovery
MLD messages	Required for Multicast operations

Note: Each IPv6 specific ICMP feature is in **bold**, each required ICMP feature is in *italics*.

9.3 Securing Autoconfiguration

The current operational practice in an IPv4 environment is to somehow keep track of which machine is using which IPv4 address at a certain point in time either by statically allocating IPv4 addresses or by using DHCP and keeping lease logs. It is also very common to identify machines in the enterprise network management systems by their MAC addresses. It is thus crucial for the secure and efficient operation of IPv4 networks to log the IP address, MAC address and L2 port combinations. There are also some tools, implemented in L2 switches, to prevent DHCP abuse and ARP poisoning called DHCP snooping and ARP inspection respectively. Let’s see what is possible in IPv6, and what countermeasures are possible to prevent abuse. There are different possible ways to assign IPv6 addresses as described in the following sections.

9.3.1 Using Stateless Address Autoconfiguration

In this method the globally aggregatable unicast address is derived from the prefix advertised by the routers and the IEEE EUI-64 identifier (RFC 2462). Since the EUI-64 identifier is generated from the MAC address if it was available, then mapping the IP address to a MAC address is very easy to do. Only MAC addresses and L2 port mapping should be implemented. Enforcing the usage of the EUI-64 identifiers as part of IPv6 addresses could be easily enforced by firewalls. Some firewalls already allow checks MAC address and EUI-64 address consistency of the outgoing packets. This way the accountability of the outgoing communications can be easily provided. However, you should configure carefully your firewall rules if you also use statically assigned addresses.

9.3.2 Using Privacy Extensions for Stateless Address Autoconfiguration

Addresses of this type were developed due to concerns that the same Interface identifier could be used anytime in multiple communication contexts. In this case it becomes possible for that identifier to be used to correlate seemingly unrelated activity. But privacy extended addresses are considered harmful [DS04] for several reasons:

- They complicate debugging, troubleshooting
- They require frequent updates on the reverse DNS entries
- They allow easier in-prefix address spoofing
- In the current form temporary and forged addresses cannot be distinguished
- They do not improve the prefix privacy

Therefore we do not recommend using privacy extended address as defined in RFC 3041. The updated standard addresses [NDK05] solve some of the problems above. There is also a new IPv6 feature called Cryptographically Generated Addresses (CGA) [RFC3972], which generates a random interface identifier based on the public key of the node. The goal of CGA is to prove ownership of an address and to prevent spoofing and stealing of existing IPv6 addresses.

To prevent using RFC 3041 type of addresses you can use the filtering technique described in the previous section.

9.3.3 Using DHCPv6

DHCPv6 is the "statefull address autoconfiguration protocol" and the "statefull autoconfiguration protocol" referred to in "IPv6 Stateless Address Autoconfiguration" (RFC2461).

DHCP can provide a device with addresses assigned by a DHCP server and other configuration information, which are carried in options.

DHCPv6 (RFC 3315) servers use DUIDs (DHCP Unique Identifier) to identify clients for the selection of configuration parameters and in association with IA clients (Identity association - a collection of addresses assigned to a client). DHCP clients use DUIDs to identify a server in messages where a server needs to be identified.

The DUID can be generated from several different sources:

1. DUID Based on Link-layer Address Plus Time (DUID-LLT)
2. DUID Assigned by Vendor Based on Enterprise Number (DUID-EN)
3. DUID Based on Link-layer Address (DUID-LL)

In the case of DUID-LLT and DUID-LL, the association between the IPv6 address and Link-layer address (usually MAC) still exist in the state information of the DHCPv6 server, so accountability is still possible. In the case of DUID-EN it is the responsibility of the administrator to build such a pairing.

If the addresses are assigned from a well identifiable sub-range in /64 the firewalls can ensure that only hosts using DHCPv6 for address configuration can connect outside of the protected network.

Unfortunately, currently there is no similar technique available on the market that will allow only real DHCPv6 servers to assign addresses to the requester hosts.

9.3.4 Static Address Assignment

The static address assignment is very similar to IPv4 static assignment therefore similar pitfalls might possible if it is used.

9.3.5 Prevention techniques

A technique similar to the one that prevents ARP cache poisoning (in IPv6 ND cache poisoning) is possible but it requires DHCPv6 snooping. Firewalls can enforce the DHCPv6 usage and make the DHCPv6 address assignment the default method, thus making DHCPv6 snooping easier to implement. Currently no DHCPv6 snooping support is available for any networking device.

IPv6 can provide an option to prevent ND cache poisoning in the case of stateless autoconfiguration via snooping the Neighbour Solicitation and Neighbour Advertisement messages: Neighbour Solicitation messages contain an informational pair [source_IPv6, source_MAC] that can be stored, while Neighbour Advertisement messages contain two informational pairs: [source_IPv6, source_MAC] and [destination_IPv6, destination_MAC] which can be also stored. Any case of a mismatch can be diagnosed from the previously stored ND entry and the switch can disable the abusing port. A "light version" of the above protocol can be implemented in Firewalls: detect and report ND entry changes i.e. different IP address with same MAC address etc.

9.3.6 Fake router advertisements

Router Advertisements are one of the well-known differences between IPv6 and IPv4. IPv4's common method to supply an address for a (default) gateway is either through DHCP or static configuration. In IPv6, geographic network routers that are connected to the same link may use the Neighbour Discovery protocol for a variety of purposes, such as discover each other's presence, determine each other's link-layer addresses, learn parameter values necessary for communicating and exchange information about prefixes they know about. However, such mechanism has a cost in terms of risk from the security viewpoint. The potential range of attacks that one could make taking the place of a network segment's default gateway is considerable.

Routers consider the information carried in router advertisements sent by other on-link routers as authoritative, even though such information is not cryptographically secured (e.g., digitally signed or key-MACed or encrypted). Therefore, routers update the affected communication parameters accordingly, without any verification. In the absence of any verification of the received information, malicious nodes may inject bogus values for optional fields of the ICMPv6 extension header, such as the advertised prefix, link layer address or MTU. Since legal router advertisements do not necessarily carry values for all of the possible options defined by the actual state of the Neighbour Discovery protocol, there is a good chances that optional values proposed by malicious router advertisements are not be corrected by successive legitimate router advertisements. As an example, if a malicious router advertisement announces an MTU of 17 bytes and legal router advertisements do not specify the MTU option, the MTU value will remain 17 until a later router advertisement, either legal or fake, announces a different value.

A similar case applies to fields such as current hop limit and reachable time, which can be exploited since they allow the sender to leave their value “unspecified”. In this case the receiver continues using its current values for those parameters. Thus, if the current value had been set via a fake router advertisement message, followed by a sequence of legitimate router advertisements that did not specify any value for the parameters, the bogus values would be used continuously until an explicit change occurs, if ever. Since parameters such as retransmission time, current hop limit and reachable time are seldom changed once they have been set, an attacker can easily poison the network. IPv6 service could thus degrade (by generation of extra hops) or become inaccessible. Network administrators should be aware of this phenomenon, avoiding configurations where such advertisements are configured by default. The use of DHCPv6 systems may also assist in preventing such rogue configurations. While this problem doesn’t represent a major threat, it can reduce end-user confidence about IPv6 services.

When a "fake router" starts to divert traffic, it will probably operate as an "evil proxy", modifying contents of outbound packets, or acting as the end-node on a communication stream. These two types of attacks can be mitigated using the IPSec protocol, whenever possible. Without knowing the keys of a specific end-to-end communication, there is no point in diverting it or intercepting it, except for DoS purposes.

But IPSec may not be an option if one end of the communication is not known in advance, if there are a large number of peers, or they are located in a different management domain. Once again, using DHCPv6, may provide the extra level of control needed to reduce advertisement problems.

A possible counter measure that system/network administrators can deploy could be a mechanism that queries ff02::2 constantly in order to identify any "alien router" on the network segment. This type of solution is not an ideal one because it can only warn about an anomaly, not really being able to prevent or correct it. But correctly diagnosing a problem is half way to solve it.

Another (weak) solution would be to set up the "real router(s)’" advertisements settings in such a way that they force themselves as preferred paths on the end-nodes. However, any serious attempt intended to hack a network segment will certainly have this possibility also embedded in its design.

Of course, all the types of attacks (hijacking, DoS, DDoS, etc.) using fake router advertisements will only be possible after an intruder compromises one node on the same segment their other targets are located.

9.4 IPv4-IPv6 Co-existence Specific Issues

A lot of work has been undertaken inside the 6NET project to investigate and report on the existing IPv6 mechanisms. This section looks closely on the potential risks deploying the mechanisms and reports on the security issues raised by the use of them with the overall aim to create awareness to the people that manage the migration to an IPv6 network.

The next sections reviews general issues arising by the use of tunnels especially automatic tunnels and operational issues of NAT-PT

Generally, any form of tunnelling poses a security threat to a network. If set up properly tunnels can effectively circumvent and undermine any security features present to guard the network like access control lists and firewalls. In a way they drill a hole through them since these security measures only "see" the outer layer of the packets, which might be well within the permitted parameters but have nothing at all to do with the contents/protocol/traffic inside. So if this traffic reaches a tunnel end-point inside the guarded network it is decapsulated and from there can potentially be very harmful since within a network itself, defence levels are usually much lower. Tunnels used for IPv6 deployment are no exception.

During the migration from IPv4 to IPv6 three different kinds of tunnels may be used: IPv6-in-IPv4, IP(v4)-in-IPv6 or other layer tunnels. In terms of general management of tunnels, RFC 4087 [RFC4087] describes managed objects used for managing tunnels of any type over IPv4 and IPv6 networks.

9.4.1 General Management Issues with Tunnels

Often the tunnel MTU is not specifically configured on the end-points when a tunnel is set up. The system then chooses a default MTU. In Cisco's IOS, for example, the default MTU will be derived from the interface MTZ of the interface towards the other end-point (by subtracting the encapsulation overhead). Even if this happens to result in the same MTU at both end-points at the time the tunnel is set up, the MTUs may diverge later (e.g. if one of the end-points starts to support jumbo frames on the egress interface, or routing changes cause the egress interface to move to one with a different MTU).

The resulting situation is a logical IP(v6) subnet where not all interfaces have the same MTU. This violates a fundamental assumption in IP networking and causes connectivity problems.

However, the symptoms are such that this situation is often hard to diagnose. In the direction from the end-point with the smaller MTU towards the receiver with a larger MTU interface, no problems will show but the other way around packets with larger size than the receiver's MTU will usually be ignored and counted as errors at the receiver. Typical tests with ping or traceroute will not show any problems because these tools use small packets. Protocols like BGP may work over the tunnel most of the time but may come to the point where the side with larger MTU must send a large amount of data and uses larger packets than the other side can take.

Unfortunately some devices (seen on Cisco routers under IOS) ignore attempts to configure a 1480-byte MTU on a tunnel towards a 1500-byte MTU interface, because this is already the default. In these cases we recommend to fix the devices so that they still accept the manually configures MTU. This will guard against problems arising when the "default MTU" changes.

MTU incompatibilities are detected by some routing protocols such as OSPFv3, which is very useful for debugging. However, such protocols usually are not used over inter-domain tunnels, where problems are most likely to occur. From this point of view it would seem useful if BGP-4 had an option to advertise link MTU in the single-hop case. Alternatively, MTU validation could be made part of a link liveliness detection protocol such as BFD (bidirectional forwarding detection).

Note that path MTU discovery ([RFC1191, [MHL05]) would make it possible for the end-points to discover the largest MTU that can be supported by the underlying network without fragmentation, but this doesn't solve the inconsistent MTU problem, because there is no guarantee that the path MTUs in both directions end up being the same. Also it isn't always implemented for tunnel interfaces (see IPv6-in-IPv4 tunnels).

9.4.2 IPv6-in-IPv4 tunnels

This category of tunnels covers the most basic and well-known transition mechanisms Manually Configured Tunnels, 6to4 and ISATAP. We will cover any specific operational, management and security issues of these mechanisms below. All of these mechanisms however have quite a few issues in common since they all encapsulate IPv6 packets in IPv4 packets.

9.4.2.1 General security issues with IPv6-in-IPv4 tunnels

Security Issues with IPv6-in-IPv4 tunnels in general lie mainly in the above mentioned problem of these tunnels circumventing and subverting security measures present for IPv4, specifically normal (IPv4-based/IPv6 unaware) firewalls on which IPv4-encapsulated IPv6 traffic only registers as IP protocol type 41 (IPv6). If one wants to use IPv6-in-IPv4 tunnels, this protocol type has to be permitted in the firewall's rules and in some cases also protocol type 58 (ICMPv6). This will effectively open a hole in the carefully configured and maintained security of the site as the traffic is let through without further inspection. This IPv6 traffic can be anything and, if the tunnel end-point also acts as an IPv6 router and forwards IPv6 traffic inside the site's IPv6 network, upon reaching the tunnel end-point inside the site could go anywhere undetected after decapsulation (though the potential damage is in most cases limited to the broadcast domains that the tunnel end-point resides in).

> *An example:*
>
> A site filters all incoming and outgoing (IPv4) http traffic unless it originates from or goes to a specific host (proxy). This protects otherwise unprotected web servers inside the network (i.e. web interfaces for configuration of network components) against attacks from the outside. The other way around this could (if the proxy were properly configured) prevent access to certain websites from inside the site. If the site now uses any IPv6-in-IPv4 tunnel mechanism to get (global) IPv6-connectivity, this tunnel most likely needs to pass the firewall to an end-point on the inside, which then becomes the site's IPv6 border router. Any (IPv6) http traffic may then travel anywhere from and to IPv6 nodes in the network, which again leaves IPv6 enabled (and connected) network nodes with IPv6 enabled web interfaces vulnerable to attacks from the outside, if they are executed via IPv6.

The best way to remedy this problem is to install an IPv6 capable firewall on the tunnel end-point that examines and properly filters the incoming IPv6 traffic after it has been decapsulated from the IPv4 wrapping. This firewall could mirror any rules present for IPv4 at the site's border for IPv6. This is the only way to let only specific IPv6 traffic in and out of the site through the tunnel.

If one only wants to filter specific traffic, one could theoretically employ bitwise filtering and look for specific bit patterns in the payload of the packets. This might make it possible to filter on at least IPv6 source and destination addresses; but this is very tiresome, prone to mistakes and not at all scalable. We do not assume that anyone would try to do this but want to state specifically that we recommended not using this method nor any other kind of packet filtering that will not work on the decapsulated IPv6 packets.

Even when using a proper IPv6 firewall on the decapsulated packets, one must be careful when setting up a tunnel because any host could potentially spoof the other end-point's IPv4 address and send IPv6-in-IPv4 encapsulated packets. The local tunnel end-point will not know that the source of these packets is not the real remote tunnel end-point and decapsulate the IPv6 traffic which can then (if not further inspected/filtered) freely enter the local IPv6 network. The attacker does not even need to know the IPv6 addresses being used on this network as it can send an ICMPv6 packet to all hosts on the tunnel link using its own IPv6 global address and retrieve the IPv6 addresses used in the network. This problem is not easily solved and in its essence is not really specific to IPv6-in-IPv4 tunnelling because it is based on IPv4 address spoofing. The best way to ward against this is doing strict RPF checking at the site's edge but even then, one cannot be completely sure. One more or less relies on other sites filtering packets with IPv4 source addresses that are not from their site at their edge preventing them from being sent out to the Internet. Concerning IPv6 the local tunnel end-point can add some additional security by dropping packets that turn out to be link-local after decapsulation. This is not always possible though, since some services or protocols rely on the use of link-local (unicast or multicast) addresses. These protocols (e.g. PIM, RIPng) can potentially be attacked by anyone. If encryption or authentication facilities are available for these services they should be used. For the other services, no real filtering can be done. We have already seen the example of a broken IPv6 multicast network because an attacker was sending bad PIM announcements, causing a bad PIM topology on the tunnel end-point. Even if a protocol run over the tunnel is not using link-local addresses (like BGP) the implementation of authentication/encryption is advised.

9.4.2.2 General management issues with IPv6-in-IPv4 tunnels

Management of IPv6-in-IPv4 tunnels depends more on the specific mechanism used to set up the tunnel(s). However, concerning monitoring, these tunnels have in common, that after configuration they behave like a point-to-point link and will only appear as one hop concerning pings and traceroutes. Unfortunately this “link-like” behaviour does not extend to features like notifications when a link goes down or up, as one can see them with real physical links. One will only recognize a tunnel going down by the fact that packets can no longer be transmitted but debugging and finding the reason for the failure is much harder, and needs to be performed by hand on the “IPv4 way” the tunnel takes, which of course may vary without anybody noticing. Therefore both for maintenance as well as of course performance reasons IPv6-in-IPv4 tunnels should only be set up over topologically short IPv4 distances

Since IPv6-in-IPv4 tunnels are purely IP they cannot be used for routing protocols like IS-IS. They can however be used for BGP without problems.

The MTU for a tunnel link is less than the path MTU for the tunnel since an IPv4 header must be added to all packets going through the tunnel. In general this might lead to undesired fragmentation effects. For IPv6 the use of path MTU discovery makes this a much smaller problem, but it is not always implemented (for tunnels). Some IPv6 implementations instead just always use the minimal IPv6 MTU without checking before. The minimal MTU is 1280 for IPv6-in-IPv4 encapsulated packets. This will avoid the problem of fragmentation in nearly all cases, since the IPv4 path MTU is often at least 1500 at the cost of adding unnecessary overhead when a larger MTU would be possible.

There might be IPv6 implementations that do not allow the same management operations for tunnel interfaces as for physical interfaces. We have seen at least one implementation that did not allow tcpdump on tunnel interfaces. There are probably other examples.

Purely hardware based routers will need specialised hardware to be able to encapsulate and decapsulate packets so that they can be used as tunnel end-points. Routers that do some operations in hardware and some in software will probably be able to handle this. One should be aware, though, that the software processing power might be limited, and the CPU used for the processing is probably also used for other tasks.

9.4.2.3 Manually configured IPv6-in-IPv4 tunnels

Other than the above mentioned general security and management issues for IPv6-in-IPv4 tunnels there are no specific problems with manually configured tunnels. Out of all transition mechanisms building upon these kinds of tunnels, manually configured tunnels however are considered to be the most stable and operationally secure due to the high level of control the administrator has over them. On the other hand, they do require the most work upon setup and both IPv4 addresses are hardcoded into the configuration which makes it impossible to use these kinds of tunnels between end-points with dynamic IPv4 addresses (i.e. over dial-in lines), at least without some kind of extra automatic setup procedure which we cover in a separate section on Tunnel Brokers.

We have already seen one implementation of tunnels that did not check if the source address of the IPv4 packet was the one configured by the administrator. Any host could potentially send IPv6 packets through the tunnel. It is always recommended to at least check, if the IPv4 source address of an IPv6-in-IPv4 packet is the IPv4 address of the other end-point, even if this doesn’t guard against spoofed packets.

9.4.3 6to4

Special issues with 6to4 mainly relate to the way IPv6-in-IPv4 tunnels are set up automatically and the security issues arising when somebody operates a (public) 6to4 relay. For the following section a 6to4 host or router is a host with just a 6to4 pseudo interface. This host might or might not have native IPv6 connectivity. Similarly it might or might not announce its 6to4 prefix to a subnet and thereby act as IPv6 access/default router for this subnet. A 6to4 relay is a dual-stack host with a 6to4 pseudo interface, which forwards packets between the 6to4 domain (2002::/16) and the rest of the IPv6 Internet. A 6to4 relay is also a 6to4 host.

In terms of management and security a network for which a 6to4 host acts as a border router is not affected by the fact that the border router uses 6to4 to provide outside connectivity (either globally or just within the 6to4 domain) aside from the fact that this network’s IPv6 connectivity of course depends on the 6to4 connectivity of the 6to4 host.

9.4.3.1 Security issues with 6to4

6to4 hosts or routers accept and decapsulate IPv4 traffic from anywhere. Constraints on the embedded IPv6 packets or where IPv4 traffic is automatically tunnelled to are minimal. Two kinds of attacks are therefore possible:

1. The 6to4 pseudo-interface can be attacked remotely with tunnelled link-local packets. If the interface is not insulated from the host’s other interfaces (which is rarely the case in practice) attacks like this could result in a corrupted neighbour cache for the whole system.

   This threat can be averted by adding an access list to the pseudo-interface to filter out bad tunnelled packets:

   - deny from 2002::/16 to 2002::/16
   - allow from 2002::/16 to 2000::/3
   - deny everything else

2. As stated above 6to4 hosts decapsulate and possibly forward any traffic coming in to the pseudo interface. They cannot distinguish between malicious IPv4-encapsulated IPv6 traffic and valid traffic coming from 6to4 relays. This “functionality” can be used both for unidirectional source address spoofing and the reflection of Denial-of-Service attacks against native IPv6 nodes. The latter is not a very big problem since the traffic can not be multiplied and might even be adversely affected by going through bottlenecks like 6to4 relays,

decapsulation and encapsulation. The only problem here is, that an attacker can more easily cover his tracks. The unidirectional source address spoofing of course also exists without 6to4 but becomes harder because the attacker needs to know a valid (existing) IPv6 address. This is a lot easier with 6to4 present because here the attacker can just take any non-6to4 address.

Attacks like these two can also only be remedied by employing sufficient filters. For example all IPv6 nodes inside the site can be guarded from attacks, if the 6to4 pseudo interface does not accept traffic from the IPv6 prefix(es) used inside the site. This also means that the site's own 6to4 prefix should be filtered on input.

Additional security issues with 6to4 relays are due to the fact that 6to4 relays by nature have a native IPv6 connection in addition to IPv4 and relay rather freely between the two. Native IPv6 nodes anywhere can use the relay as a means to obscure their identity when attacking (possibly even IPv4 nodes). Attackers from IPv6 can attack IPv4 hosts with tunnelled packets sending spoofed 6to4 packets via a relay to the IPv4 hosts. The relay can obscure identity, if it relays any packets whilst not checking if the 6to4 address actually matches the IPv4 host the packet comes from. Note that for relays it is assumed that it is at least configured in a way as to not relay between different 6to4 addresses (except of course from or to other known 6to4 relays), thereby facilitating IPv4 to IPv4 attacks.

1. A 6to4 relay can be used for locally directed (IPv4) broadcast attacks. For example if the relay has an interface with address w.x.y.z/24 an attacker could send packets with a 6to4 address that translates into the address w.x.y.255. This is even possible to remote locations if "no ip directed broadcast" is not configured.

   This problem however is easily remedied by another entry in the access list, which prevents packets with destination similar to the above 6to4 address from getting in.

2. The issue mentioned above is actually only a special case of the general problem of 6to4 relays becoming a part of DoS attacks against IPv4 nodes which might be totally unaware of 6to4 but get hit by encapsulated packets nevertheless. If the attack further is executed with a spoofed source address (which is easily possible as stated above) the source of the attacks cannot be traced. A 6to4 relay can also be used for address spoofing and therefore anonymization of attacks coming from native IPv6 hosts

Generally, a 6to4 relay can be reasonably well protected if the validity of source or destination 6to4 addresses is always checked. That is, it should be checked if the enclosed IPv4 address is a valid global unicast IPv4 address. It could even be restricted to only accepting and forwarding 6to4 encapsulated traffic where the 6to4 destination or source address matches the actual IPv4 address the packets come in from or go to. As with the general rule about no forwarding between 6to4 addresses however, exceptions must be made for traffic coming from or going to known other 6to4 relays.

For more information about security considerations with 6to4 please refer to [RFC3964].

9.4.3.2 Management issues with 6to4

However well protected a 6to4 relay may be, the traffic going through should always be monitored, especially if the relay is configured with the well-known IPv4 anycast address for public 6to4 relays.

Other than monitoring no particular management is required for 6to4 since it was specifically designed for ease of use and low maintenance.

9.4.4 ISATAP

ISATAP is another automatic tunnelling mechanism based on the automatic creation of IPv6-in-IPv4 tunnels. As such – from a security point of view – it should not be used, if manually configured IPv6-in-IPv4 tunnels are an option. However, since ISATAP is specifically meant to be used only within a site and if correspondingly protected, it is a reasonably secure and low maintenance mechanism, to provide isolated dual-stack hosts with IPv6 connectivity to the site's main IPv6 network and thereby global IPv6 connectivity.

9.4.4.1 Security issues with ISATAP Clients and Servers

An ISATAP server or router should be protected in such a way as to only permit incoming tunnels from the hosts inside the site. This can be accomplished with simple IPv4 firewall rules. Additionally the site's normal IPv4 border router should permit incoming and outgoing protocol 41 (IPv4 encapsulated IPv6 traffic) only for source and destination addresses belonging to known tunnels. This is not only to protect the ISATAP servers but all ISATAP clients in the site as well, as all clients connected to the same ISATAP server are essentially on the same (IPv6) link and cannot be easily protected from one another.

If the list of ISATAP servers is in any way made automatically available via DNS, DHCP or other means it should be very well maintained.

Since ISATAP clients and servers perform actual neighbour discovery when first starting to communicate with the only difference being that the ISATAP routers do not send unsolicited router advertisements, the same procedures to secure neighbour discovery should be taken as in any native IPv6 network.

9.4.4.2 Management issues with ISATAP

Monitoring traffic between the ISATAP hosts at a site is difficult. All hosts using the same ISATAP router are on the same virtual link, so the packets do not really pass through any other routers (of course the packets might pass through IPv4 routers on the layer below but there they are hardly distinguishable from the normal IPv4 traffic). Monitoring of non-link-local traffic can thus really only be done on the ISATAP router but itself. Note that ISATAP clients within the site can send packets to each other directly using IPv6-in-IPv4 encapsulation and their link-local ISATAP addresses. This traffic does not go through the ISATAP server and can only be monitored at the sending and receiving nodes which is hardly feasible for all hosts of the site.

9.4.5 Teredo

Teredo (also known as IPv4 network address translator (NAT) traversal for IPv6) is designed to make IPv6 available to IPv4 hosts through one or more layers of NAT by tunnelling packets over UDP. It is a host-to-host automatic tunnelling mechanism that provides IPv6 connectivity, when dual-stack hosts are located behind one ore multiple NATs by encapsulating IPv6 packets in IPv4-based User Datagram Protocol (UDP) messages.

9.4.5.1 Security Considerations for Teredo

The threats posed by Teredo can be grouped into four different categories:

1. Opening a hole in the NAT
2. Using the Teredo service for a man-in-the-middle attack

3. DoS of the Teredo Service
4. DoS against non-Teredo nodes

These four types of threats as well as possible mitigating strategies are addressed below.

Opening a Hole in the NAT

Teredo is designed to make a machine reachable via IPv6 through one or more layers of NAT. That means that the machine which uses the service consequently gives up any firewall service that was available in the NAT box. All services opened for local use will become potential targets for attacks from the entire IPv6 Internet. It is recommended to use a personal (IPv6) firewall solution, i.e. a piece of software that performs the kind of inspection and filtering locally that is otherwise performed in a perimeter firewall as well as the usage of IPv6 security services such as IKE, AH, or ESP. Since Windows XP Teredo clients are most common these days, we would like to point out at this point that Windows XP (since SP2 or the advanced networking pack) comes with an acceptable IPv6 firewall.

Man-in-the-Middle Attacks

The goal of the Teredo service is to provide hosts located behind a NAT with a globally reachable IPv6 address. There is a possible class of attacks against this service in which an attacker somehow intercepts the router solicitation, responds with a spoofed router advertisement and provides a Teredo client with an incorrect address. The attacker may have one of two objectives: a) it may try to deny service to the Teredo client by providing it with an address that is in fact unreachable, or b) it may try to insert itself as a relay for all client communications, effectively executing a man-in-the-middle attack. It is not possible to use IPv6 security mechanisms such as AH or ESP to prevent these kinds of attacks since they cover only the encapsulated IPv6 packet but not the encapsulating IPv4 and UDP header. In fact it is very hard to find an effective signature scheme to prevent such an attack since the attacker does not do anything else than what the NAT legally does. The Teredo client should systematically try to encrypt outgoing IPv6 traffic using IPSec. That will at least make spoofing of the IPv6 packets impossible and prevent third parties from listening in to the communication. By providing each client with a global IPv6 address Teredo enables the use of IPSec.

Denial of the Teredo Service by Server Spoofing or an Attack of the Servers

Spoofed router advertisements can be used to insert an attacker in the middle of a Teredo conversation. The spoofed router advertisements can also be used to provide a client with an incorrect address pointing to either a nonexistent IPv4 address or to the IPv4 address of a third party. The Teredo client will detect the attack when it fails to receive traffic through the newly acquired IPv6 address of the so-called Teredo server. Using authentication this attack can be prevented.

Other than confusing clients with false server addresses the Teredo service can of course also be disrupted by mounting a Denial of Service attack against the real Teredo servers and relays sending a huge number of packets in a very short time. Since Teredo servers are generally designed to handle quite a large amount of network traffic this attack most likely will have to be quite brute force, if it should work at all. The attack is mitigated if the Teredo service is built redundantly and the clients are ready to “failover” to another server. That will of course cause the clients to renumber.

If a Teredo relay is attacked in such a way it should stop announcing the reachability of the Teredo service prefix to the IPv6 network. The traffic will be picked up by the next relay.

Denial of Service against non-Teredo Nodes

There is a widely expressed concern that transition mechanisms such as Teredo can be used to mount denial of service attacks by injecting traffic at locations where it is not expected. These attacks fall into three categories: a) using the Teredo server as a reflector in a denial of service attack, b) using the Teredo server to carry a denial of service attack against IPv6 nodes and c) using the Teredo relays to carry a denial of service attack against IPv4 nodes. A common mitigating factor in all of these cases is the “regularity” of the Teredo traffic which contains highly specific patterns such as the Teredo UDP port or the Teredo IPv6 prefix. In cases of attacks these patterns can be used to quickly install filters and remove the offending traffic.

9.4.6 GRE Tunnels

The use of IPv4 GRE (Generic Route Encapsulation) tunnels provides another means to transport IPv6 over an IPv4-only network. In most cases they are used because unlike IPv6-in-IPv4 tunnels where IPv6 is directly encapsulated in IPv4 datagrams GRE can be used for the Intermediate System to Intermediate System (IS-IS) routing protocol.

9.4.6.1 Security issues with GRE tunnels

If GRE tunnels are to go through an IPv4 firewall this firewall has to be opened for IP protocol type 47 for IPv4 datagrams coming from or going to the remote tunnel end-point.

GRE tunnel end-points are authenticated by a simple key that is transmitted during the setup of the tunnel. Since the key is transmitted in clear text format this doesn’t really add much security and the key is also not used for encryption of any kind.

9.4.6.2 Management of GRE tunnels

The broader functionality of GRE tunnels comes at the cost of an even shorter MTU, since the GRE header also has to be included in each packet. Other than that, GRE tunnels can be managed like IPv6-in-IPv4 tunnels or point-to-point links respectvely.

9.4.7 OpenVPN Tunnels

OpenVPN is (as the name indicates) a VPN solution. Licensed under the GPL it creates cross platform (layer 2) point-to-point or Ethernet-bridge tunnels over which IPv6 can easily be transported. The software multiplexes IPv6 in IPv4 UDP packets using functionality provided by the OpenSSL library, which may optionally be encrypted. As a VPN solution one has of course to regard one end of the tunnel as the "server" end and one as the client but both ends use the same software. The tunnel end at the site that provides IPv6 connectivity acts as the server. For more information on how OpenVPN works, please refer to the project’s homepage at: http://openvpn.sourcefourge.net

9.4.7.1 Security Issues with OpenVPN tunnels

In terms of security OpenVPN has the great advantage of providing authenticated and optionally even encrypted tunnels. It is based on OpenSSL for certification and either uses static pre-shared keys or TLS for dynamic key exchange. The use of X.509 certificates can be regarded as very secure. It can only be compromised, if the secret key is not kept safe.

The certificates are not bound to specific hosts. They can be used anywhere between any two hosts. So an owner of a certificate could put both public and private key on his laptop and with that set up an authenticated tunnel from anywhere where he has IPv4 connectivity. This, of course, is the desired functionality for any Virtual Private Network solution but in comparison to the usual IPv6-in-IPv4 tunnels this has quite a few advantages for the deployment of IPv6 on for example dial-in lines where users not usually have static IPv4 addresses. It provides the user with much more flexibility at the cost of security relying solely on the fact that the user keeps his keys safe and only uses them for himself.

9.4.7.2 Management Issues with OpenVPN tunnels

OpenVPN tunnels are very robust and work even on rather unstable/unreliable IPv4 connections between both end-points. They are known to survive even ISDN or DSL reconnects where the client comes back with a different IPv4 address. In this case just a new TLS handshake is performed to authenticate both sides and the tunnel is back online.

In and of itself the mechanism is not automated but it is an ideal basis for setting up a tunnel broker:

- The use of a CA enables a centralized management of access authorization and trust.
- Failure of the tunnel broker's hardware or the IPv4 link between tunnel broker client and server does not impose administrative work other than fixing hardware or link. The service continues seamlessly after the IPv4 link between client and server is re-established. The FQDN is used to identify a server and hence DNS entries may be changed to redirect tunnel broker clients to a working server in the case of a failure.
- The persistence of the IPv6 link is very good because of mechanisms inherent to the OpenVPN software.
- OpenVPN traverses most NATs without the need of additional configuration. If the NAT does not support this traversal, fowarding of a single UDP port to the OpenVPN client suffices to establish connectivity.

9.4.8 Dual-stack

Dual-stack is the conceptually easiest and for quite some time to come the best way of deploying IPv6. The drawback to this scenario is the fact that it involves the maintenance of two separate IP infrastructures including management and security.

9.4.8.1 Security considerations for dual-stack networks or hosts

The most important paradigm for security in dual-stack networks or on stand-alone dual-stack hosts is that (if this network or host is also provided with global IPv6 connectivity) security for every IPv6 host must mirror exactly the security provisions in place for IPv4. Every firewall rule and every access list that is restricting access to a host needs to be "translated" into corresponding rules and access lists for IPv6. This is not always easy, especially if the network topology is not the same for IPv6 and IPv4. In that case access lists and firewall rule sets cannot be mirrored at all but need to be composed in such a way that they culminate in the exact same level of security for IPv6 for every host as for IPv4.

A special case is, when there's not even global IPv4 connectivity in a network, because that network sits behind a NAT and is addressed with private addresses. For IPv6 on the other hand all hosts could be addressed with globally unique (and reachable/routed) addresses, if connectivity is for example provided through a tunnel. In this case security for IPv6 needs to be designed from scratch although present firewall rules for the NAT itself can provide a basis, if they are translated to corresponding IPv6 rules.

9.4.8.2 Management (and performance) issues with dual-stack networks

An important aspect of dual-stack deployment is performance. Dual-stack hosts are configured to always prefer IPv6 when a hostname resolves into both an A (IPv4 address) and an AAAA (IPv6 address) record. The deployment of dual-stack services (e.g. FTP mirror) with different performance for IPv4 and IPv6 must be avoided because the IP layer does not remain transparent. We have seen the deployment of a dual-stack FTP mirror with a poor IPv6 performance, causing all the people used to upgrade their applications on this mirror having deployed IPv6 FTP clients to get a 80kBps bandwidth instead of 4Mbps for their downloads. This issue can of course only be controlled for services one deploys oneself. For remote services the only thing one can do is to keep the reason for these problems in mind (and to educate unknown users accordingly). It is important that people know that this does not happen because IPv6 is slower in and of itself.

One further management issue in deploying an IPv6/IPv4 dual-stack network lies in configuring both internal and external routing for both protocols. If one has for example used OSPFv2 for intra site routing before, adding IPv6 to the Layer 3 network one will either make the transition to OSPFv3 or IS-IS necessary or one will at least be forced to run one of these IGPs in addition to OSPFv2.

9.4.9 DSTM

DSTM (Dual Stack Transition Mechanism) is a tunnelling solution for IPv6-only networks, where IPv4 applications are still needed on dual-stack hosts within an IPv6-only infrastructure. IPv4 traffic is tunnelled over the IPv6-only domain until it reaches an IPv6/IPv4 gateway, which is in charge of packet encapsulation/decapsulation and forwarding between the IPv6-only and IPv4-only domains. The solution proposed by DSTM is transparent to any type of IPv4 application and allows the use of layer 3 security.

9.4.9.1 Security Considerations with DSTM

The DSTM mechanism can use all of the defined security specifications for each functional part of its operation. E.g. for DNS, the DNS Security Extensions/Update can be used.

Concerning address allocation, when connections are initiated by the DSTM nodes, the risk of Denial of Service attacks (DoS) based on address pool exhaustion is limited in the intranet scenario. With the intranet scenario, if DHCPv6 is deployed, the DHCPv6 Authentication Message can be used for security. When using TSP for address allocation, the SSL encryption and authentication can be used since TSP messages are in plain text.

When exchanging the DSTM options using DHCPv6, the DSTM Global IPv4 Address option may be used by an intruding DHCP server to assign an invalid IPv4-mapped address to a DHCPv6 client in a denial of service attack. The DSTM Tunnel Endpoint option may be used by an intruding DHCP server to configure a DHCPv6 client with an endpoint that would cause the client to route packets through an intruder system. To avoid these security hazards, a DHCPv6 client must use authentication to confirm that it is exchanging the DSTM options with an authorized DHCPv6 server. The DSTM Ports option may be used by an intruding DHCP server to assign an invalid port range to a DHCP client in a denial of service attack. To avoid this security hazard, a DHCP client must use authenticated DHCP to confirm that it is exchanging the DSTM options with an authorized DHCP server.

The main difference between the intranet scenario and the VPN scenario of DSTM is security. In the VPN scenario, DHCPv6 must not be used for address allocation but TSP (tunnel set up protocol) with SSL encryption can be used for this purpose.

In the VPN scenario, the DSTM server must authenticate the outside DSTM client. This authentication cannot rely on the IPv6 address since the address depends on the visiting network but can be based on some shared secret.

In the VPN scenario, the mapping between the IPv4 and the IPv6 address of the DSTM node in the TEP is also a security concern. If the mapping is established dynamically (no configuration by the DSTM server), it could be possible for every intruder knowing a valid temporary IPv4 address to use the TEP as an IPv4 relay or to access internal IPv4 resources. So, in the VPN scenario, the mapping in the TEP must be managed by the DSTM server which authenticates the DSTM host and its IPv6 address. This is an important requirement that avoids the use of IPv4 resources by non authorized nodes.

Finally, for IPv4 communications on DSTM nodes, once the node has an IPv4 address, IPSec can be used since DSTM does not break secure end-to-end communications at any point. The tunnel between the DSTM host and the TEP can be ciphered, but it is our view that this is more of an IPv6 feature (like the use of IPv6 mobility) than a DSTM feature

9.4.10 NAT-PT/NAPT-PT

As noted in RFC 2766 [RFC2766], NAT-PT and end-to-end security do not work together. When an IPv6-only node (X) initiates communication to IPv4-only node Y, the packets from X have certain IPv6 source and destination addresses which are both used in IPSec (AH or ESP) and TCP/UDP/ICMP checksum computations. Since NAT-PT translates the IPv6 address of X into an IPv4 address that has no relationship to X’s IPv6 address, there is no way for recipient Y to determine X’s IPv6 address and in that way verify the checksums.

9.4.10.1 Prefix Assignment

RFC2766 does not explain how the IPv6 nodes learn about the prefix that is used to route packets to the NAT-PT box. If the prefix is pre-configured in IPv6 nodes, the IPv6 node would prepend the preconfigured prefix to the address of any IPv4-only node with which it wants to initiate communications. However, with a prefix, there might be a reachability problem if the NAT-PT box were to shut down unexpectedly. If an attacker would somehow be able to give the IPv6 node a fake prefix, the attacker would be able to steal all of the node’s outbound packets to IPv4 nodes.

Even though this is not specified in RFC 2766, DNS servers and DNS-ALGs should be used for outgoing connections to return the prefix information to the IPv6 node as a means to avoid the problem of a statically preconfigured prefix. When an IPv6-only node wishes to initiate communications with an IPv4-only node, its resolver would send an AAAA query. This query can be passed through the DNS-ALG which itself looks for an A record. In this case the DNS-ALG can prepend the appropriate prefix for NAT-PT itself and thus return a full AAAA record to the IPv6-only node.

9.4.10.2 Security Issues Arising when Using a DNS-ALG

A DNS-ALG is required when IPv4-only nodes should be allowed to initiate communication within a NAT-PT scenario. Since the DNS-ALG will translate simple “A record” requests into “AAAA record” requests and vice versa DNSSEC will not work in this case. However, as pointed out in draft-durand-v6ops-natpt-dns-alg-issues [Dur03], if the host sets the “AD is secure” bit in the DNS header, it is possible for the local DNS server to verify signatures. Also another option to increase security is for the DNS-ALG to verify the received records, translate them and sign the translated records anew. A third option would be if the host had an IPSec security association with the DNS-ALG to protect DNS records.

In case the DNS-ALG also monitors the state of a number of NAT-PT boxes and use only the prefixes of those that are running. The method by which a DNS-ALG determines the state and validity of a NAT-PT box must of course also be secure. The DNS-ALG and each NAT-PT box should be configured with a pairwise unique key that will be used for integrity-protected communications. Note that messages from a DNS-ALG are not integrity-protected and can therefore be modified. To prevent such a modification, a DNS-ALG can sign its packets. The DNS-ALG’s public key can be made available like that of any other DNS server (see RFC 2535 [RFC2535]) or presented form of a certificate that has a well known root CA. A shared key technique may not be as practical.

9.4.10.3 Source address spoofing attack

There are two cases in which an attacker will use NAT-PT resources, one where the attacker is in the same stub domain as the NAT-PT box and the second where the attacker is outside the NAT-PT stub domain.

Suppose that an attacker is in the same stub domain as the NAT-PT box and sends a packet destined for an IPv4-only node to the other side of the NAT-PT-gateway, forging its source address to be an address that topologically would be located inside the stub domain. If the attacker sends many such packets, each with a different source address, then the pool of IPv4 addresses may quickly get used up, resulting in a DoS attack (or rather Address depletion attack). A possible solution to this attack as well as to similar attacks like resource exhaustion or a multicast attack is to perform ingress filtering on the NAT-PT box (which is the border router). This would prevent an attacking node in its stub domain from forging its source address and thus from performing a reflection attack on other nodes in the same stub domain. The NAT-PT box should also drop packets whose IPv6 source address is a multicast address. Address Depletion attacks can be prevented by employing NAT-PT in a way that it translates the TCP/UDP ports of IPv6 nodes into the corresponding TCP/UDP ports of the IPv4 nodes/addresses. However, sessions initiated by IPv4 nodes are restricted to one service per server. Of course IPSec might be used to further increase security.

Suppose now that an attacker outside the NAT-PT domain sends a packet destined to an IPv6-only node inside the NAT-PT domain and forges its (IPv4) source address to be an address from the IPv4 address pool used for NAT-PT. The same attacks are then possible as in the scenario above. Again filtering can be used to prevent this. The NAT-PT gateway should drop all packets whose IPv4 source address is a broadcast/multicast address. It should also filter out packets from outside that claim to have a source address from inside the NAT-PT domain.

9.4.11 Bump in the API (BIA)

Security issues with BIA mostly correspond to those of NAT-PT. The only difference is that with BIA address translation occurs in the API and not the network layer. The advantage here is that, since the mechanism uses the API translator at the socket API level, hosts can utilise the security of the underlying network layer (e.g. IPSec) when they communicate via BIA with IPv6 hosts using IPv4 applications.

Another security issue NAT-PT and BIA have in common stems from the use of address pooling, which may open a denial of service attack vulnerability. One should employ the same sort of protection techniques as mentioned fore NAT-PT in this regard.

Note that since there is no DNS ALG necessary with BIA as it is with NAT-PT, there is no interference with DNSSEC when using this transition mechanism.