IPv6 and IPv4 Threat Comparison and Best-Practice Evaluation (v1.0)

Sean Convery (sean@cisco.com)
Darrin Miller (dmiller@cisco.com)

Table of Contents

1 Introduction 2
1.1 Caveats 3
2 Overview of IPv4 Topology and Best-Practice Security Rules 3
3 Threat Analysis 4
3.1 Attacks with New Considerations in IPv6 4
3.1.1 Reconnaissance 4
3.1.2 Unauthorized Access 7
3.1.3 Header Manipulation and Fragmentation 11
3.1.4 Layer 3-Layer 4 Spoofing 13
3.1.5 ARP and DHCP Attacks 15
3.1.6 Broadcast Amplification Attacks (smurf) 16
3.1.7 Routing Attacks 17
3.1.8 Viruses and Worms 18
3.1.9 Translation, Transition, and Tunneling Mechanisms 19
3.2 Attacks with Strong IPv4 and IPv6 Similarities 20
3.2.1 Sniffing 20
3.2.2 Application Layer Attacks 21
3.2.3 Rogue Devices 21
3.2.4 Man-in-the-Middle Attacks 21
3.2.5 Flooding 21
4 Overview of IPv6 Topology and Best-Practice Security Rules 22
5 Summary 25
6 Acknowledgments 25
7 Change Log 26
8 References 26
Appendix A: Current and Future Directions for Research 28
Appendix B: Configurations from the Lab 29

1 Introduction

IPv6 [1] security is in many ways the same as IPv4 [2] security. The basic mechanisms for transporting packets across the network stay mostly unchanged, and the upper-layer protocols that transport the actual application data are mostly unaffected. However, because IPv6 mandates the inclusion of IP Security (IPsec) [3], it has often been stated that IPv6 is more secure than IPv4. Although this may be true in an ideal environment with well-coded applications, a robust identity infrastructure, and efficient key management, in reality the same problems that plague IPv4 IPsec deployment will affect IPv6 IPsec deployment. Therefore, IPv6 is usually deployed without cryptographic protections of any kind. Additionally, because most security breaches occur at the *application* level, even the successful deployment of IPsec with IPv6 does not guarantee any additional security for those attacks beyond the valuable ability to determine the source of the attack.

Some significant differences, however, exist between IPv4 and IPv6 beyond the mandate of IPsec. These differences change the types of attacks IPv6 networks are likely to see. It is also unlikely that the average organization will migrate completely to IPv6 in a short timeframe; rather it will likely maintain IPv4 connectivity throughout the multiyear migration to IPv6. To date, however, there has not been a thorough treatment of the threats such networks will face and the design modifications needed to address these threats.

This paper outlines many of the common known threats against IPv4 and then compares and contrasts how these threats, or similar ones, might affect an IPv6 network. Some new threats specific to IPv6 are also considered. The current capabilities of available products are evaluated, as is how any inherent protocol characteristics of IPv6 affect the nature of the threat. This is prefaced by a brief overview of current best practices around the design of an IPv4 Internet edge network and then followed by a review of how that IPv4 edge network needs to evolve in order to secure the addition of IPv6.

The appendices of this document highlight the configurations used in the test lab in support of this paper and point to areas of future research. For several points in this paper, reasonable hypotheses are identified but no actual testing has yet been performed. Over time it is the hope of the authors that the community can contribute to the testing of these areas and their results integrated into this document.

This document is meant to benefit the following groups of individuals:

- Network and security architects—This large body of individuals has been responsible for building the Internet today and has remained, with the exception of certain countries, largely disengaged from the IPv6 protocol and its modifications. By reviewing this document, these architects should be able to apply the concepts discussed here to their own areas of the Internet to ensure that as IPv6 is deployed it is done in a secure manner from the outset rather than slowly migrating toward security, as happened with IPv4.
- Security researchers—Reading this document should stimulate further ideas for research in IPv6 security; some ideas are defined in Appendix A.
- IETF members—The IETF [4], the group responsible for the development of the IP protocol, should benefit from a comparative study of the threats in IPv4 as compared to IPv6.
- Government policy makers—The U.S. Department of Defense has stated that it plans a full migration to IPv6 by 2008 [5], spurred in part by its goal for security. Although this goal is admirable, IPv6 is not a panacea for all security problems. Additionally, a substantial investment in the development of new training materials for government employees will be required to meet the 2008 deadline. Furthermore, other groups inside the government have focused on IPv6 as a means to improve Internet security. This document should help such groups identify areas that need attention.

1.1 Caveats

IPv6 security is a large and complex subject. It is also one that has seen little examination, except by the group who designed the protocol themselves. Therefore, some topics are not addressed in this document. For example, this document does not address Mobile IP Version 6 (MIPv6) [6], which is still in the draft stage in the IETF. Some of the implications regarding the support of the routing header (a key element in MIPv6) are discussed, but only as the routing header impacts a static IPv6 network.

Additionally, this document focuses on the security requirements of medium to large edge networks on the Internet. These networks typically house some element of public services (Domain Name System [DNS], HTTP, Simple Mail Transfer Protocol [SMTP]) and a filtering router or firewall protecting their internal resources. The document does not address the implications of the threats to service providers (or other core network entities).

Finally, because of the ubiquity of their deployment, Cisco routers are the principal network entity tested in this research. The threats and mitigation techniques described in this document should apply to a network built with any vendor's equipment, however, and the configurations provided should be easily modified as necessary.

2 Overview of IPv4 Topology and Best-Practice Security Rules

Figure 1 shows a typical IPv4 Internet edge network.

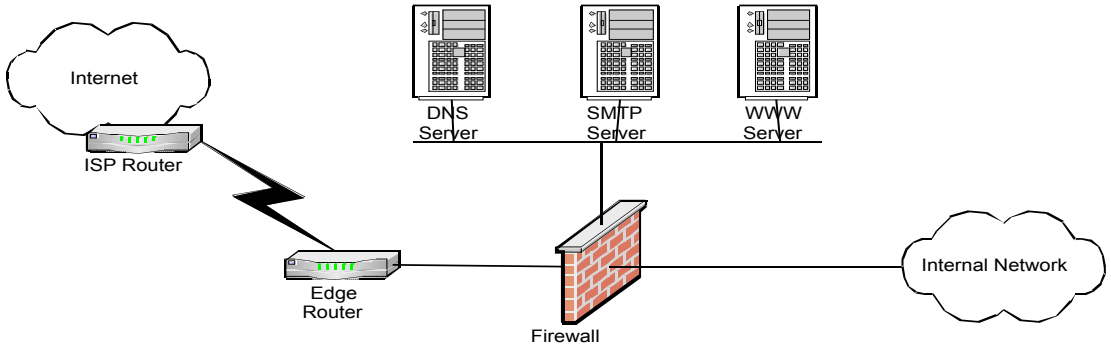

Figure 1 Typical IPv4 Internet Edge Network

In this network, two principal points of network security enforcement are inline: the firewall and the edge router. These devices can be combined into a single device in some cases, or they can have additional security technologies applied to them directly or to companion devices not shown in the figure. These include intrusion detection systems (IDSs), application proxies, and so on. Additionally, each of the public servers can have host security controls such as antivirus software, host intrusion detection, file system integrity checkers, host firewalls, and so on.

This basic design and its countless variations are in use today in thousands of networks around the world. The mechanisms to provide security to networks designed this way are well-understood, as are the limitations of this approach. In support of this paper, a network was built in the lab emulating this design and configurations; and diagrams are provided in Appendix B. The specific techniques used to secure this network for IPv4 threats are summarized in section 3 as each threat is explored.

It should be noted that for larger organizations it is becoming increasingly difficult to identify perimeters within a network. The introduction of IPv6 (or any new core technology) could make things even more perilous without an adequate understanding of the threats identified in section 3.

3 Threat Analysis

This section evaluates and compares threats in IPv4 and in IPv6. It is divided into two main sections, the first of which outlines attacks that significantly change as a result of IPv6, and the second summarizes attacks that do not fundamentally change.

3.1 Attacks with New Considerations in IPv6

The following nine attacks have substantial differences when moved to an IPv6 world. In some cases the attacks are easier, in some cases more difficult, and in others only the method changes.

- Reconnaissance
- Unauthorized access
- Header manipulation and fragmentation
- Layer 3 and Layer 4 spoofing
- Address Resolution Protocol (ARP) and Dynamic Host Configuration Protocol (DHCP) attacks
- Broadcast amplification attacks (smurf)
- Routing attacks
- Viruses and worms
- Transition, translation, and tunneling mechanisms

3.1.1 Reconnaissance

The first category of attack is reconnaissance, which also is generally the first attack executed by an adversary. In this attack the adversary attempts to learn as much as possible about the victim network. This includes both active network methods such as scanning as well as more passive data mining such as through search engines or public documents. The active network methods have the goal of giving the adversary specific information about the hosts and network devices used in the victim network, their interconnections with one another, and any avenues of attack that can be theorized based on the evaluation of this data.

3.1.1.1 IPv4 Considerations

In IPv4 the adversary has several well-established methods of collecting this information:

- Ping sweeps–By determining the IPv4 addresses in use at an organization (through active probes, whois lookups, and educated guesses), an adversary can systematically sweep a network with ICMP or Layer 4 ”ping” messages that solicit a reply, assuming both query and response are not filtered at the network border. Following this scan, the adversary uses the data to formulate some hypothesis regarding the layout of the victim network. Tools such as traceroute and firewalk can provide further data to aid the adversary.
- Port scans–After identifying reachable systems, the adversary can systematically probe these systems on any number of Layer 4 ports to find services both active and reachable. By discovering hosts with active services, the adversary can then move to the next phase.

- Application and vulnerability scans–The adversary can then probe these active ports by various means to determine the operating system and the version numbers of applications running on the hosts, and even test for the presence of certain well-known vulnerabilities.

Some tools such as Nmap [7] can perform elements of all these scan types at the same time.

Attack mitigation techniques for these reconnaissance techniques are generally limited to filtering certain types of messages used by an adversary to identify the resources of the victim network and trying to detect the reconnaissance activity that must be permitted. Reconnaissance activity cannot be stopped completely because the very act of permitting communications with your devices permits some form of reconnaissance.

3.1.1.2 IPv6 Considerations

This section outlines the differences in the reconnaissance attack when moved to IPv6. Because port and application vulnerability scans are identical after a valid address is identified, this section focuses on identifying valid addresses. The first subsection highlights technology differences independent of currently available technology, and the latter outlines current capabilities in this area for the adversary and the defender.

3.1.1.2.1 Technology and Threat Differences

With regard to technology, IPv6 reconnaissance is different from IPv4 reconnaissance in two major ways. The first is that the ping sweep or port scan, when used to enumerate the hosts on a subnet, are much more difficult to complete in an IPv6 network. The second is that new multicast addresses in IPv6 enable an adversary to find a certain set of key systems (routers, Network Time Protocol [NTP] servers, and so on) more easily. Beyond these two differences, reconnaissance techniques in IPv6 are the same as in IPv4. Additionally, IPv6 networks are even more dependent on ICMPv6 to function properly. Aggressive filtering of ICMPv6 can have negative effects on network functions. ICMPv6 filtering alternatives are reviewed in section 3.1.2.

3.1.1.2.1.1 IPv6 Subnet Size Differences

The default subnet size of an IPv6 subnet is 64 bits, or 2^{64}, versus the most common subnet size in IPv4 of 8 bits, or 2^{8}. This increases the scan size to check each host on a subnet by $2^{64} - 2^{8}$ (approximately 18 quintillion). Additionally, the 64-bit address is derived based on the EUI-64 version of a host MAC address, or in the case of IPv6 privacy extensions [8] (which are enabled by default in Windows XP and available on numerous other platforms), the number is pseudorandom and changes regularly. So a network that ordinarily required only the sending of 256 probes now requires sending more than 18 quintillion probes to cover an entire subnet. Even if we assume that sound network design principles are discounted and that the same 64-bit subnet now contains 10,000 hosts, that still means only one in every 1.8 quadrillion addresses is actually occupied (assuming a uniform random distribution). And even at a scan rate of 1 million probes per second (more than 400 Mbps of traffic), it would take more than 28 years of constant scanning to find the first active host, assuming the first success occurs after iterating through 50 percent of the first 1.8 quadrillion addresses. If we assume a more typical subnet with 100 active hosts, that number jumps to more than 28 centuries of constant 1-million-packet-per-second scanning to find that first host on that first subnet of the victim network.

Now it should be noted that many variables can make this scanning easier for the adversary. First, public services on the Internet edge need to be reachable with DNS, giving the adversary at least a small number of critical hosts within the victim network to attack. Second, the large nature of IPv6 addresses and the lack of a strict requirement for Network Address Translation (NAT) will cause more networks to adopt dynamic DNS or other mechanisms to ensure that even hosts have a valid DNS name (typing FE80:CA01:0:56::ABCD:EF12:3456 to talk to your friend's PC is no fun). This means that a compromise

of a DNS server within the organization under attack could yield large caches of hosts. Third, administrators may opt for easy-to-remember host addresses for key systems (::10,::20,::F00D, and so on) that could be entered into a database used by the scanning tool. These easy-to-remember names could include simply mapping the decimal v4 last octet to the hex v6 last octet, because dual stack will be the norm for years to come (this was done in the lab detailed in Appendix B for convenience). Fourth, by focusing on popular IEEE OUI designations for NIC vendors, an adversary could significantly reduce the number space of 2^{64}. And finally, by exploiting poorly secured routers or other gateway devices, an adversary could view the IPv6 neighbor-discovery cache data (the functional equivalent of an ARP cache) to find available hosts, or could simply turn on a packet-capture capability such as tcpdump to find addresses available to scan.

Also, like in IPv4 networks, the internal hosts should be protected by a firewall that limits or completely prevents uninitiated conversations from reaching these systems.

The implications of these larger subnets are significant. Today's network management systems often use ping sweeps as a method of enumerating a network for an administrator. New techniques need to be adopted for this purpose (perhaps neighbor cache checks on routers). Based on initial testing, the neighbor cache is populated on a router only when the device is communicated with by the router (such as sending off-net traffic).

Additionally, this has potentially far-reaching implications for the way Internet worms are propagated, whether they are random address-based or use some form of hierarchical address designations. The basic assumption is that worms will have a much more difficult time propagating in the same manner as they have in IPv4. This is an area that requires further research; some ideas are highlighted in Appendix A.

3.1.1.2.1.2 New Multicast Addresses

IPv6 supports new multicast addresses that can enable an adversary to identify key resources on a network and then attack them. These addresses have a node, link, or site-specific domain of use as defined in RFC 2375 [9]. For example, all routers (FF05::2) and all DHCP servers (FF05::3) have a site-specific address. Although this setup clearly has a legitimate use, it is in effect handing the adversary an official list of systems to further attack with simple flooding attacks or something more sophisticated designed to subvert the device. Therefore, it becomes critical that these internal-use addresses are filtered at the border and not reachable from the outside.

3.1.1.2.2 Current Technology Capabilities

Today there is no known ping sweep tool for IPv6. Nmap, which supports ping sweeping in v4, elected not to support it in IPv6, most likely for the reasons outlined in section 3.1.1.2.1.1. On the detection side, some IDS systems today (host or network) do not support IPv6, making detection of the scanning activity difficult. This will improve as more vendors ship IPv6 inspection capabilities. Current versions of most popular network firewalls do support IPv6, meaning that filtering various messages to complicate the reconnaissance efforts of the adversary is possible.

On the network management side, very few–if any–network management tools have been developed to deal with the host identification problem outlined in this section.

3.1.1.3 Candidate Best Practices

Based on the changes in reconnaissance attacks in IPv6, the following candidate best practices are suggested:

- *Implement privacy extensions carefully*–Although privacy extensions are a benefit from an obscurity standpoint regarding scanning attacks, they can also make it difficult to trace problems and troubleshoot issues on a network. If a network has a misbehaving host and that host's address changes regularly, it could be quite difficult to trace the exact host or to determine if the problems are

from one host or many. Better options are to use static addresses for internal communication that are MAC address-based and pseudorandom addresses for traffic destined for the Internet. In addition, this makes current audit capabilities to track worms more challenging because when we track an infection back to a particular subnet, the privacy extensions rotation of the addresses or a machine reboot could make it difficult to identify the infected end host.

- *Filter internal-use IPv6 addresses at organization border routers*—Administrators can define site-local addresses for their organization, including specific multicast addresses such as the all-routers address FF05::2. These site-local addresses can potentially lead to new avenues of attack, so administrators must filter these addresses at the organization's border routers.
- *Use standard, but nonobvious static addresses for critical systems*—Instead of standardizing on host addresses such as ::10 or ::20, try something that is more difficult for adversaries to guess, such as ::DEF1 for default gateways. This is certainly a "security through obscurity" technique, but because it involves little additional effort on the administrator's part, its use has no drawbacks. The goal here is to make it difficult for the adversary to guess the global addresses of key systems. Standardizing on a short, fixed pattern for interfaces that should not be directly accessed from the outside allows for a short filter list at the border routers.
- *Filter unneeded services at the firewall*—Like in IPv4, your public and internal systems should not be reachable on services that they do not need to be reached on. Though some are hoping that tools such as IPsec will eliminate the need for firewalls, they will be around for years to come as Layer 3 and 4 filtering is well understood. Until some nontechnical issues (such as the international politics of who controls any trust roots) are resolved, wide-scale deployment of IPsec will be impractical for both IPv4 and IPv6.
- *Selectively filter ICMP*—Because neighbor discovery uses ICMP and fragmentation is done only on end stations (which requires path maximum-transmission-unit discovery [PMTUD]), it is imperative that some ICMP messages be permitted in IPv6. That said, nonessential ICMP messages can be filtered at a firewall, as can ICMP echo and echo-reply messages, if that aspect of manageability can be sacrificed. The author's recommend that, particularly for IPv6, ICMP echo be enabled in all directions for all hosts, except that inbound ICMP echoes from the Internet to the internal network should be denied. Additionally, IPv6 requires ICMPv6 neighbor discovery-neighbor solicitation (ND-NS) and neighbor discovery-neighbor advertisement (ND-NA) messages to function (described in section 3.1.2), as well as router-solicitation (RS) and router-advertisement (RA) messages if autoconfiguration is used and RA messages are sent from the router for prefix lifetime advertisements. Finally, as in IPv4, packet-too-big messages should be broadly permitted to ensure proper functioning of PMTUD. Section 3.1.2.2.1.3 describes the ICMP messages required in more detail.
- *Maintain host and application security*—Although timely patching and host lockdown are critical elements in IPv4, they are even more critical during the early stages of IPv6 because many host protections (firewalls, IDSs, and so on) do not yet broadly support IPv6. Additionally, it is highly likely (though testing is necessary; refer to Appendix A) that the initial introduction of IPv6 into networks will result in some hosts not being properly secured. It is necessary to focus on maintaining host security to ensure that hosts that are compromised will not become stepping stones to compromise other end hosts.

3.1.2 Unauthorized Access

Unauthorized access refers to the class of attacks where the adversary is trying to exploit the open transport policy inherent in the IPv4 protocol. Nothing in the IP protocol stack limits the set of hosts that can establish connectivity to another host on an IP network. Attackers rely upon this fact to establish connectivity to upper-layer protocols and applications on internetworking devices and end hosts.

3.1.2.1 IPv4 Considerations

IPv4 networks have concentrated on limiting unauthorized access by deploying access control technologies within the end systems and on gateway devices in between the IPv4 endpoints. These controls can occur at both Layer 3 and Layer 4. The access control methods in IPv4 get more complex as you move up the protocol stack. At the IP layer, the defender uses basic access control lists (ACLs) to allow only approved hosts to send packets to a device. The ACLs are intended to limit access to or through a device based on security policy and by doing so, limit the available avenues of attack to specific services available on the network. In IPv4 networks, these access controls are implemented in networking devices (firewalls) and on end devices themselves (host firewalls). Although firewalls can implement security policy based on information in the IPv4 headers only, they are best used when combined with upper-layer inspection of TCP/UDP and application layer information.

3.1.2.2 IPv6 Considerations

The need for access control technologies is the same in IPv6 as in IPv4, though eventually the requirement for IPsec may enable easier host access control. The defender wants to limit the ability of the adversary to gain avenues of attack against services on an end host. The ability to do access control based in IPv6 changes not only the information that can be filtered in the Layer 3 header, but also the way the addressing and routing systems of IPv6 are architected. The addressing system of IPv6 changes from that for IPv4 because it includes the ability for one adapter in an IPv6-enabled node to have multiple IPv6 addresses. These multiple IPv6 addresses have significance for communicating on the local subnet (link local - FE80::/10), within an organization (site local – FC00::/16 or FD00::/16 pending working group decision), or on the Internet at large (global unicast addresses – aggregates of prefix binary 001). When the use of these address ranges is combined with the routing system, the network designer can limit access to IPv6 end nodes through IPv6 addressing and routing.

For instance, with IPv6 the network designer can assign global unicast addresses only to devices that need to communicate with the global Internet while assigning site-local addresses to devices that need to communicate only within the organization. Likewise, if a device needs to communicate only within a particular subnet, only the link-local address is needed. Additionally, the use of IPv6 privacy extensions, as mentioned earlier, can limit the time any single IPv6 address is accessible and exposed to a security threat. Beyond the previously stated differences in IPv6, the following sections outline the differences in the unauthorized access attack avenues when the network moves to IPv6. The first subsection highlights the technology differences in the IPv4 and IPv6 header that are independent of currently available technology, and the latter outlines current capabilities in this area for the adversary and the defender.

3.1.2.2.1 Technology and Threat Differences

In IPv6 the basic function of mitigating access to other IP devices based on policy is still implemented with firewalling and ACLs on end hosts and internetworking devices. However, numerous significant differences between the IPv6 and IPv4 headers may change how an administrator deploys these technologies. The following paragraphs discuss some of the areas of difference.

3.1.2.2.1.1 IPsec

When implemented with IPv4 or IPv6, IPsec has similar impacts on the administrator's ability to enforce security policy with IP header information. The following discussion points apply to both IPv4 and IPv6. If IPsec encryption is implemented from end to end, current firewalling technology is effective only in applying policy based on Layer 3 information because of the cryptographic protections. If IPv6 uses only the authentication header, it is conceivable that IPv6-capable firewalls could inspect the upper-layer protocols within the authentication-header (AH) encapsulation and permit or deny access to the packet based on that information.

3.1.2.2.1.2 Extension Headers

IP options in IPv4 are replaced with extension headers in IPv6. With this replacement, extension headers may be used in an attempt to circumvent security policy. For example, all IPv6 endpoints are required to accept IPv6 packets with a routing header. It is possible that in addition to *accepting* IPv6 packets with routing headers, end hosts also *process* routing headers and forward the packet. With this possibility, routing headers can be used to circumvent security policy implemented on filtering devices such as firewalls [10].

To avoid this possibility, the network manager should designate the specific set of nodes that are to act as MIPv6 home agents (typically the default router for the subnet). The network designer should also validate that the operating systems within their organization do not forward packets that include the routing header. If operating systems that do forward packets that include the routing header are on the network, then the network designer must configure the network to filter the routing header on access control devices. If MIPv6 is not needed, packets with the routing header can be easily dropped at access control devices without relying on the end host to not forward the packets. Although it is easy to start with a "no MIPv6" policy, the emerging applications on handheld devices with WiFi access will make that stance challenging to maintain. For this reason it is best to make sure the end system policy is correctly implemented as "no-forwarding."

3.1.2.2.1.3 ICMP

ICMPv6 is an integral part of IPv6 operations, even more so than in IPv4. Current best practice for IPv4 firewalling of ICMP is sometimes debated, but it is generally accepted that stringent ICMP filtering is a best practice. In some extreme cases all ICMP messages should be filtered. This blanket prohibitive filtering is simply not possible in IPv6. For the purposes of this document, comparing and contrasting how a generic ICMPv4 policy would translate to ICMPv6 is critical. The following ICMPv4 messages are permitted through the firewall, and all others are denied. The general rules are to permit these inbound ICMP messages from the Internet to a DMZ on a firewall and deny ICMP to the firewall device. These rules may be more or less stringent than a given administrator's ICMP policy, but are included here only for the sake of demonstration.

- ICMPv4 Type 0 - echo reply
- ICMPv4 Type 3 Code 0 - Destination unreachable net unreachable
- ICMPv4 Type 3 Code 4 – Fragmentation needed but don't-fragment (DF) bit set
- ICMPv4 Type 8 - Echo request
- ICMPv4 Type 11 - Time exceeded

In contrast, an ICMPv6 firewall policy needs to support additional messages not only through the device but also to and from the firewall device.

ICMPv6 messages required to support equivalent functions to the firewall policy stated previously are as follows:

- ICMPv6 Type 1 Code 0 – No route to destination
- ICMPv6 Type 3 - Time exceeded
- ICMPv6 Type 128 and Type 129 - Echo request and echo reply

New IPv6 messages potentially required to be supported through the firewall device follow:

- ICMPv6 Type 2 - Packet too big–This is required for PMTUD to function correctly because intermediate nodes on an IPv6 network are not allowed to fragment packets. Though allowing PMTUD to function in IPv4 is useful, in IPv6 intermediary devices cannot fragment, so this message becomes more critical to proper network operations.

- ICMPv6 Type 4 - Parameter problem–This is required as an informational message if an IPv6 node cannot complete the processing of a packet because it has a problem identifying a field in the IPv6 header or in an extension header. Further research into the potential abuse of this message type is needed.

ICMPv6 messages potentially required to be supported to and from the firewall device are as follows:

- ICMPv6 Type 2 – Packet too big–The firewall device must be able to generate these messages for proper MTU discovery to take place, because the firewall device cannot fragment IPv6 packets.
- ICMP Type 130-132 - Multicast listener messages–In IPv4, IGMP would need to be permitted for multicast to function properly. In IPv6 a routing device must accept these messages to participate in multicast routing.
- ICMPv6 Type 133/134 – Router solicitation and router advertisement–These are necessary for a variety of reasons, most notably IPv6 end-node autoconfiguration.
- ICMPv6 Type 135/136 – Neighbor solicitation and neighbor advertisement–These messages are used for duplicate address detection and Layer 2 (Ethernet MAC)-to-IPv6 address resolution.
- ICMPv6 Type 4 – Parameter problem–Refer to the previous explanation; this message may be required, but further research is warranted.

3.1.2.2.1.4 Multicast Inspection

Currently most IPv4 firewalls do minimal multicast inspection and filtering. Local-use multicast is integral to the functioning of IPv6. Firewall devices, at a minimum, need to allow the link-local multicast addresses to the firewall in order to provide neighbor discovery. Firewalls in Layer 3 mode should never forward link-layer multicasts. Devices acting as firewalls should inspect all source IPv6 addresses and filter any packets with a multicast source address.

3.1.2.2.1.5 Anycast Inspection

Additionally, although anycast as per RFC 2373 [11] is restricted to routers at this time, operating systems have started to add anycast support to their kernels. This could make anycast usage for services such as DNS or NTP [12] more prevalent in the short term. If this happens, any stateful device (firewall, network IDS [NIDS], server load balancing [SLB]) needs to make feature enhancements to its code to be able to designate an anycast address for inspection and origin servers that listen and respond to the anycast address. If this is done, then when a server that is serving an anycast service answers with its real address the stateful device can map the return traffic to the inbound-initiated traffic with the anycast address. Finally, as has been noted in [13], using IPsec and Internet Key Exchange (IKE) to secure anycast communications has limitations. Work within the IETF is ongoing, but this requirement can potentially be addressed with the use of Group Domain of Interpretation (GDOI) [14].

3.1.2.2.1.6 Transparent Firewalls

Several “Layer 2” or “transparent” firewalls on the market act as bridges while enforcing Layer 3 to Layer 7 policy. In current IPv4 networks, these devices have to be specially programmed to deal with a variety of IP and data link layer interactions such as ARP inspection and DHCPv4. In IPv6 these types of firewalls need to enhance their inspection capabilities to inspect the appropriate IPv6 ICMP and multicast messages. As discussed earlier, ICMPv6 is integral to the proper functioning of an IPv6 network, and a transparent firewall must be able to track the ICMPv6 messages that deal with neighbor discovery, duplicate address detections, autoconfiguration, and multicast management, just to name a few. These capabilities would offer a way to mitigate against attacks that spoof IP-to-MAC address bindings or spoofed DHCP messages. Refer to section 3.1.5 for more discussion on this topic. Additionally, security policy needs to be explicitly defined for the extensive use of multicast addresses in IPv6. For instance, a

bridge must forward all FF02:: multicast in IPv6. An IPv6 transparent firewall must be able to define filters to forward the link local all multicast nodes (FF02::1) address that is used in IPv6 functions such as autoconfiguration.

3.1.2.2.2 Current Technology Capabilities

Though many IPv6-capable firewalls are available, many are implementing partial solutions for IPv6 for time-to-market reasons. For example, some IPv6 firewalls understand only a subset of the extension headers in IPv6, and they drop IPv6 traffic that includes these headers. An example is a firewall that does not have logic to process the routing header. If the firewall receives a packet with the routing header, it discards the packet. This behavior has some security benefit when the firewall is protecting hosts that might unpack and forward a packet with a routing header. However, this behavior precludes the firewall from being utilized in an environment that requires MIPv6.

3.1.2.3 Candidate Best Practices

Based on the differences in the IPv6 header and associated extension headers, the following candidate best practices are suggested:

- *Determine what extension headers will be allowed through the access control device*–Network designers should match their IPv6 policy to their IPv4 IP options policy. If any IPv4 IP options are denied on the access control device, the IPv6 access control device should implement the same policies. Additionally, administrators should understand the behavior of the end-host operating system when dealing with the extension headers and dictate security policy based on that behavior. For instance, as noted earlier, the administrator should validate that end-host operating systems do not forward packets that contain a routing header.
- *Determine which ICMPv6 messages are required*–It is recommended that administrators match their policy map closely to the equivalent ICMPv4 policy with the following additions:
  - ICMPv6 Type 2 - Packet too big
  - ICMPv6 Type 4 – Parameter problem
  - ICMPv6 Type 130-132 – Multicast listener
  - ICMPv6 Type 133/134 – Router solicitation and router advertisement
  - ICMPv6 Type 135/136 – Neighbor solicitation and neighbor advertisement

3.1.3 Header Manipulation and Fragmentation

The third category of attack is fragmentation and other header manipulation attacks. This category of attack has been primarily used for one of two purposes. The first purpose is to use fragmentation as a means to evade network security devices, such as NIDS or stateful firewalls. The second purpose of the attack is to use fragmentation or other header manipulation to attack the networking infrastructure directly.

3.1.3.1 IPv4 Considerations

In IPv4 fragmentation is a technique used to fit the IPv4 datagram into the smallest MTU on the path between end hosts. IPv4 fragmentation has been used as a technique to bypass access controls on devices such as routers and firewalls. Fragmentation also has been used to obfuscate attacks in order to bypass network security monitoring products such as NIDS. Most modern firewall and NIDS products go to great lengths to reassemble packets and match the reassembled packets to access control rules or to attack signatures. In general, large amounts of fragmented traffic have been used as an early indicator of an

intrusion attempt because most baselines of Internet traffic indicate that the percentage of fragmented traffic is low [15].

3.1.3.2 IPv6 Considerations

This section outlines the differences in the fragmentation attacks when moved to IPv6. The first subsection highlights technology differences independent of currently available technology, and the latter outlines current capabilities in this area for the adversary and the defender.

3.1.3.2.1 Technology and Threat Differences

IPv6 fragmentation by intermediary devices is prohibited per RFC 2460 (refer to sections 4.5 and 5). One of the most common fragmentation attacks uses overlapping fragments to obfuscate attacks from IPv4 security devices. In IPv6, overlapping fragments is not a proper way of handling fragmentation based on the rules outlined in RFC 2460; these fragments can possibly be viewed as an attack and dropped. Additionally, if the overlapping packets are allowed to bypass the security device, several end-host operating systems drop overlapping fragments in their IPv6 stack software. However, if the end operating system does accept overlapping fragments, there is nothing to prevent the adversary from using fragmented packets in an attempt to bypass the IPv6 security device policy for similar purposes as the IPv4 fragmentation attacks. Additionally, an adversary can still use out-of-order fragments to try to bypass string signatures of a network-based IDS.

RFC 2460 section 5 says "IPv6 minimum MTU is 1280 octets." For this reason, administrators can allow the security device to drop fragments with less than 1280 octets unless the packet is the last packet in the flow. Administrators can perform this action *if* the sending operating system fragments the original packet at the MTU supplied by the PMTUD messages and continues to create this size of IPv6 fragments until the last segment of the original packet is delivered. If the host operating system does not behave in this manner, then the security device has to continue to accept and process IPv6 fragments with less than 1280 octets. This behavior would continue to allow obfuscation of attacks by sending large amounts of small fragmented packets. Baselining the fragmentation and reassembly behavior of popular operating systems is necessary to validate the potential of this filtering.

Additional fragmentation issues should be considered for devices that are not configured to do fragment reassembly (routers not running firewall), but are trying to enforce security policy based on Layer 3 and Layer 4 information. For example, in IPv4 some routers have the fragment keyword in the access control entry definition. The only packets that match this IPv4 ACL are those packets that have a fragment offset not equal to zero, that is, noninitial fragments. For IPv4 packets, we know the protocol fragments flags and offset values from the IP header, so we can easily calculate if enough of the upper-layer protocol is within the first fragment to determine the Layer 4 port number. So nonfragmented packets and first fragments go through the normal access-list process and can have the appropriate security policy applied. The combination of multiple extension headers and fragmentation in IPv6 creates the potential that the Layer 4 protocol is not included in the first packet of a fragment set, making it difficult to enforce Layer 4 policy on devices that do not do fragment reassembly. An example of this is a router running Cisco IOS Software without the firewall feature set that does fragment reassembly. With IPv6, Cisco IOS Software matches noninitial IPv6 fragments and the first fragment if the protocol cannot be determined. Cisco IOS Software also supports a new keyword "undetermined transport," which matches any IPv6 packet where the upper-layer protocol cannot be determined.

3.1.3.2.2 Current Technology Capabilities

Similar to IPv4, current IPv6 firewalls and IDSs implement fragment reassembly and other fragmentation checks in order to mitigate fragmentation attacks. These fragmentation checks include examining out-of-sequence fragments and switching these packets into order, as well as examining the number of fragments from a single IP given a unique identifier to determine denial-of-service (DoS) attacks. IPv6 has no

known fragmentation attack tools, but that does not eliminate the threat that such tools exist or can be created easily. Firewalls checking for these attacks will want to be matching on source subnets to catch the case where the adversary is using RFC 3041 addressing to generate fragment streams from what would appear to be multiple sources.

3.1.3.3 Candidate Best Practices

As stated earlier, though the handling of IPv6 fragmentation is specified to be much different than in IPv4, the threats in bypassing security devices remain the same. The following candidate best practices should be considered in IPv6 networks to limit the effectiveness of fragmentation attacks:

- *Deny IPv6 fragments destined to an internetworking device when possible*–This will limit certain attacks against the device. However, this filtering should be tested before deployment to ensure that it does not cause problems in your particular network environment.
- *Ensure adequate IPv6 fragmentation filtering capabilities*–The combination of multiple extension headers and fragmentation in IPv6 creates the potential that the Layer 4 protocol will not be included in the first packet of a fragment set. Security monitoring devices that expect to find the Layer 4 protocol need to account for this possibility and reassemble fragments.
- *Drop all fragments with less than 1280 octets (except the last one)*–RFC 2460 section 5 says "IPv6 minimum MTU is 1280 octets." For this reason security devices may be able to drop any IPv6 fragment with less than 1280 octets unless it is the last fragment in the packet. More testing is necessary in this area, as specified in section 3.1.3.2.1. A case that should be noted is for Layer 2 firewalls and IPv4 routers transporting a tunnel. There is no requirement that IPv6 packets be 1280 octets or more between Layer 3 interfaces, just that if the packet is fragmented, the fragments must be reassembled at the receiving interface before forwarding. This is done specifically to allow tunneling over IPv4 networks where the MTU might be less than 1280. In that case, IPv4 is architecturally Layer 2.

3.1.4 Layer 3-Layer 4 Spoofing

A key element enabling numerous different types of IP attacks is the ability for an adversary to modify their source IP address and the ports they are communicating on to appear as though traffic initiated from another location or another application. This so-called "spoofing" attack is prevalent despite the presence of best practices to mitigate the usefulness of the attack.

3.1.4.1 IPv4 Considerations

Today in IPv4, spoofing attacks (principally Layer 3-based) occur every day. They can make DoS, spam, and worm or virus attacks more difficult to track down. Layer 3 spoofing attacks are not generally used in interactive attacks as return traffic routes to the spoofed location, requiring the adversary to "guess" what the return traffic contains (not an easy proposition for TCP-based attacks because TCP has 32-bit sequence numbers). Layer 4 spoofing can be used in interactive attacks in order to make traffic appear to come from a location it did not (such as injecting false Simple Network Management Protocol (SNMP) messages or syslog entries). RFC 2827 [16] specifies methods to implement ingress filtering to prevent spoofed Layer 3 traffic at its origin. Unfortunately such filtering is not broadly implemented, and because it requires widespread usage to have a significant benefit, spoofed traffic is still very common. It is important to note that RFC 2827 ensures that only the network portion of an address is not spoofed, not the host portion. So in the 24-bit subnet 192.0.2.0/24, RFC 2827 filtering ensures that traffic originating from 192.0.3.0 is dropped but does not stop an adversary from spoofing all the hosts within the 192.0.2.0/24 subnet assigned to a broadcast domain. RFC 2827 does allow the administrator to track attacks to a particular organization, and tracking is one of the first steps to accountability.

In addition to stopping the spoofing of valid ranges within the IPv4 address space, a large body of addresses have not been allocated [17] in IPv4, and reserved addresses exist that will likely never be allocated [18]. These ranges can be globally blocked, and attacks that attempt to use those spoofed ranges can be identified and stopped at network choke points as implemented with a security policy.

3.1.4.2 IPv6 Considerations

This section outlines the differences in Layer 3 and Layer 4 spoofing attacks when moved to IPv6. The first subsection highlights technology differences independent of currently available technology, and the latter outlines current capabilities in this area for the adversary and the defender.

3.1.4.2.1 Technology and Threat Differences

One of the most promising benefits of IPv6 from a Layer 3 spoofing perspective is the globally aggregated nature of IPv6 addresses. Unlike IPv4, the IPv6 allocations are set up in such a way as to easily be summarized at different points in the network. This allows RFC 2827-like filtering to be put in place by Internet service providers (ISPs) to ensure that at least their own customers are not spoofing outside their own ranges. Unfortunately this is not required standard behavior, and it requires conscious implementation on the part of operators. Layer 4 spoofing attacks are not changed in any way, because Layer 4 protocols do not change in IPv6 with regard to spoofing. Just be aware that subnets are much larger in IPv6, so even with RFC 2827-like filtering an adversary can spoof an enormous range of addresses.

From a transition standpoint, the various tunneling mechanisms offer the ability for an adversary with either IPv4 or IPv6 connectivity to send traffic to the other version of IP while masking the true source. As an example, adversaries can use 6to4 relay routers to inject traffic into an IPv6 network with very little ability to trace back to the true source [19]. It should be noted that this is no worse than the inability to trace IPv4, but simple checks at the relay, such as making sure the outer IPv4 source matches the address embedded in the IPv6 source, enhances traceback from the IPv6 destination.

3.1.4.2.2 Current Technology Capabilities

Currently Layer 3 spoofing can be mitigated using the same techniques as in IPv4 with standard ACLs. Layer 4 spoofing is not changed in any way. Spoofed traffic can be detected using IPv6-capable firewalls or IDSs. Currently no techniques are available to mitigate the spoofing of the 64 bits of host address space available in IPv6. What would be useful in IPv6 networks (and IPv4 networks as well) is a method to correlate IP, MAC, and Layer 2 port pairings for traffic. This data could be stored by the switch and then polled by or sent to a management station, enabling the operator to quickly determine the physical switch port on which a given IP address is communicating.

3.1.4.3 Candidate Best Practices

Based on the changes in Layer 3 and Layer 4 spoofing attacks in IPv6, the following candidate best practices are suggested:

- *Implement RFC 2827-like filtering and encourage your ISP to do the same*–At least containing spoofed traffic to the host portion of the IPv6 address provides a large benefit for at least tracing the attack back to the originating network segment.
- *Document procedures for last-hop traceback*–With the large range of spoofable addresses in a IPv6 subnet, it is critical that when an attack does occur you have mechanisms to determine the true physical source of the traffic. This generally entails some combination of Layer 2 and Layer 3 information gleaned from switches and routers.

- *Use cryptographic protections where critical*–If an application uses strong cryptographic protections, a successful spoof attack is meaningless without also subverting the cryptographic functions on the device.

3.1.5 ARP and DHCP Attacks

ARP and DHCP attacks attempt to subvert the host initialization process or a device that a host accesses for transit. This generally involves the subversion of host bootstrap conversations through either rogue or compromised devices or spoofed communications. These attacks try to get end hosts to communicate with an unauthorized or compromised device or to be configured with incorrect network information such as default gateway, DNS server IP addresses, and so on.

3.1.5.1 IPv4 Considerations

DHCP uses a broadcast message from the client when it initially boots up, allowing a rogue DHCP server to attempt to respond to the host before the valid DHCP server is able to. This allows the rogue server to set critical connectivity settings, including default gateway and DNS server, thus enabling man-in-the-middle attacks. Additionally, DHCP messages can be spoofed, allowing an adversary to consume all available DHCP messages on the server.

ARP attacks center around spoofing ARP information to cause the IP-MAC binding of a particular host to be changed so that the IP address remains valid but the victims communicate with the adversary's MAC address. This is most often done to spoof the default gateway.

Technologies have been developed in IPv4 to address some of these attack types. For example, Cisco has a feature in Ethernet switches called DHCP snooping [20], which allows certain ports designated as "trusted" to participate in DHCP responses while most of the other ports are configured to allow sending only DHCP client messages. Additionally, a feature called ARP inspection [21] performs similar protections for ARP. Furthermore, some IDS systems can detect certain types of ARP misuse.

3.1.5.2 IPv6 Considerations

This section outlines the differences in ARP and DHCP attacks when moved to IPv6. The first subsection highlights technology differences independent of currently available technology, and the latter outlines current capabilities in this area for the adversary and the defender.

3.1.5.2.1 Technology and Threat Differences

In IPv6, unfortunately, no inherent security is added on to the IPv6 equivalents of DHCP or ARP. Because stateless autoconfiguration (a lightweight DHCP-like functionality provided in ICMPv6) can provide a viable alternative to DHCP in many cases, dedicated DHCP servers are not common in IPv6 and are not even broadly available in modern server operating systems. Dedicated DHCPv6 servers may appear in order to offer additional configuration parameters such as DNS servers, time servers, IP telephony servers, and so on, so a level of DHCP protection is still required. Unfortunately, stateless autoconfiguration messages can be spoofed, and spoofing can be used to deny access to devices. To mitigate this, the trusted port concept should be used in conjunction with router-advertised messages.

In IPv6, rather than continue with a unique version of ARP for every media type, ARP is replaced with elements of ICMPv6 called neighbor discovery. Neighbor discovery has the same inherent security as ARP in IPv4. Though the possibility of enabling some sort of more secure neighbor discovery using IPsec exists, this is far from standardized, and it involves unique implementation considerations because of the added security. The Securing Neighbor Discovery (SEND) [22] working group in the IETF is working on a solution to this problem. At present, both router and neighbor-solicitation and -advertisement messages can be spoofed and will overwrite existing neighbor-discovery cache information on a device, resulting in

the same issues present in IPv4 ARP. For instance, a spoofed router discovery could inject a bogus router address that hosts listen to and perhaps choose for their default gateway; the bogus router can record traffic and forward it through the proper routers without detection.

These ARP spoofing-like attacks have not been implemented in any publicly available test code, so some unique considerations may appear after such code is released and tested.

Although DHCPv6 is investigating security options, the protocol is too new to be considered in this paper. At a minimum the approaches used for protecting DHCP in IPv4 networks should be implemented for IPv6.

3.1.5.2.2 Current Technology Capabilities

No security tools are available today to help detect or stop DHCPv6, autoconfiguration, or neighbor-discovery abuses in IPv6. These messages can be filtered at a router or firewall like any ICMP message, but because most of these attacks are locally significant only, this will have minimal benefit. The neighbor-discovery attacks have not been implemented in any publicly available test code for IPv6, so some unique considerations may appear after such code is released and tested. Getting the equivalent inspection capability that is now present in IPv4 would help mitigate this threat.

3.1.5.3 Candidate Best Practices

Without the ability to detect the misuse of neighbor-discovery messages or to secure their transport, best practices are limited to the following:

- *Use static neighbor entries for critical systems*–In highly sensitive environments you can specify that a system has a static entry to its default router and avoid many of the typical neighbor-discovery attacks. This is a very administratively burdensome practice and should not be undertaken lightly.

3.1.6 Broadcast Amplification Attacks (smurf)

Broadcast amplification attacks, commonly referred to as “smurf” attacks, are a DoS attack tool that takes advantage of the ability to send an echo-request message with a destination address of a subnet broadcast and a spoofed source address, using the victim’s IP. All end hosts on the subnet respond to the spoofed source address and flood the victim with echo-reply messages.

3.1.6.1 IPv4 Considerations

Documented in the late 1990s, this common attack has a simple mitigation method in IPv4 networks. If IPv4-directed broadcasts are disabled on the router, when an adversary sends an echo-request message to the broadcast address of the IP subnet they end up sending one echo-reply message to the victim, as opposed to replies from all the devices on the network. According to Best Current Practice (BCP) 34 [23], the default behavior for IP routers is to turn IP-directed broadcasts off. The command `no ip directed broadcasts` is the default for Cisco IOS Software Version 12.0 and later. This specific attack is becoming less common, but can still be used to create an effective DoS attack. A current website still monitors smurf attack-capable subnets.

3.1.6.2 IPv6 Considerations

This section outlines the differences in broadcast amplification attacks when moved to IPv6. The first subsection highlights technology differences independent of currently available technology, and the latter outlines current capabilities in this area for the adversary and the defender.

3.1.6.2.1 Technology and Threat Differences

In IPv6 the concept of an IP-directed broadcast is removed from the protocol and specific language is added to the protocol designed to mitigate these types of attacks. Specifically with regard to a smurf attack, RFC 2463 [24] states that an ICMPv6 message should not be generated as a response to a packet with an IPv6 multicast destination address, a link-layer multicast address, or a link-layer broadcast address (RFC 2463 section 2.2). If end nodes are compliant to RFC 2463, then smurf and other amplification attacks used against IPv4 are not an issue in IPv6 networks.

3.1.6.2.2 Current Technology Capabilities

Our testing has shown that several popular operating systems comply with the RFC and do not respond to a echo request directed at the link-local all nodes multicast address sourced from a spoofed address. Some ambiguity still exists in the standard about whether end nodes should respond to ICMP messages with global multicast addresses as the source address. If the end nodes do respond to these multicast addresses, then an adversary could make an amplification attack on the multicast infrastructure that may cause a DoS due to resource consumption on the internetworking devices.

3.1.6.3 Candidate Best Practices

- *Implement ingress filtering of packets with IPv6 multicast source addresses*–There is no valid reason for a multicast source address, so the administrator should drop any packets with a multicast source address at the border of the network.

No other candidate best practices will be available until amplification attacks are discovered in IPv6. Specific testing needs to be performed on a range of operating system end nodes to determine their behavior when responding to an ICMP packet sourced with a global multicast address.

3.1.7 Routing Attacks

Routing attacks focus on disrupting or redirecting traffic flow in a network. This is accomplished in a variety of ways, ranging from flooding attacks, rapid announcement and removal of routes, and bogus announcement of routes. Particulars of the attacks vary, depending on the protocol being used.

3.1.7.1 IPv4 Considerations

In IPv4, routing protocols are commonly protected using cryptographic authentication to secure the routing announcements between peers. The most common implementation is a Message Digest Algorithm 5 (MD5) authentication with a preshared key between routing peers.

3.1.7.2 IPv6 Considerations

This section outlines the differences in several routing protocols underlying security mechanisms when moved to IPv6. The first subsection highlights technology differences independent of currently available technology, and the latter outlines current capabilities in this area for the adversary and the defender.

3.1.7.2.1 Technology and Threat Differences

Several protocols do not change their security mechanism when transitioning from IPv4 to IPv6.

Multiprotocol Border Gateway Protocol (BGP) was extended to carry IPv6 interdomain routing information in RFC 2545 [25]. As such, BGP continues to rely on TCP MD5 for authentication. The Intermediate System-to-Intermediate System (IS-IS) protocol [26] was extended in a draft specification [27] to support IPv6, but the extension does not change the underlying authentication of IS-IS. Originally, IS-IS provided for the authentication of link-state packets (LSPs) through the inclusion of authentication

information as part of the LSP. However, the simple password authentication was not encrypted. RFC 3567 [28] adds a cryptographic authentication to IS-IS, and this cryptographic authentication will continue to be used to protect IS-IS for IPv6 traffic.

In Open Shortest Path First Version 3 (OSPFv3) [29], the authentication fields of the OSPF header are removed. Routing Information Protocol Next-Generation (RIPng) [30] has also removed the authentication from the protocol specification. OSPF and RIPng rely on IPsec AH and Encapsulating Security Payload (ESP) headers to provide integrity, authentication, confidentiality, and antireplay protection of routing information exchanges. Additional work is being done to secure both IPv4 and IPv6 protocols, such as the "The Generalized TTL Security Mechanism" [31]. This mechanism is also applicable to IPv6-specific protocols if the Hop-Limit field in the IPv6 header is used to protect a protocol stack.

3.1.7.2.2 Current Technology Capabilities

The security mechanisms to secure protocols that have changed with IPv6, OSPFv3, and RIPng are implemented inconsistently across internetworking vendors.

3.1.7.3 Candidate Best Practices

- Use traditional authentication mechanisms on BGP and IS-IS.
- Use IPsec to secure protocols such as OSPFv3 and RIPng–This is dependant on functioning vendor implementations.

Use IPv6 hop limits to protect network devices–Investigate vendor implementations of IPv6 hop limits to protect the protocol stack from attack. For instance, a basic technique is to start the time to live (TTL) of 255 for a valid peer and ensure that the resulting TTL accepted by the router is high enough to prevent acceptance of a spoofed packet that has come from a different part of the infrastructure.

3.1.8 Viruses and Worms

Viruses and worms remain one of the most significant problems in IP networking today, with almost all of the most damaging publicly disclosed attacks in recent years having a virus or worm at its nexus.

3.1.8.1 IPv4 Considerations

In IPv4, viruses and worms not only damage the hosts themselves but also can damage the transport of the network through the increased burden to routers and mail servers around the Internet. SQL slammer [32], for example, caused massive network flooding due in part to the rate with which it scanned the network (each attack packet was a single UDP message). Timely patching, host antivirus, and early detection followed by perimeter blocking have been the three techniques used in IPv4. Early detection is most easily performed with anomaly detection systems such as those available from Arbor Networks. Additionally, newer host-based IDS products can intercept certain system calls that would have caused the compromise in the system.

3.1.8.2 IPv6 Considerations

This section outlines the differences in virus and worm attacks when moved to IPv6. The first subsection highlights technology differences independent of currently available technology, and the latter outlines current capabilities in this area for the adversary and the defender.

3.1.8.2.1 Technology and Threat Differences

A traditional virus in no way changes with IPv6. E-mail based viruses or those that infect removable media remain as you would expect. However, worms or viruses and worms that use some form of Internet scanning to find vulnerable hosts may experience significant barriers to propagation in IPv6 due to the issues raised in section 3.1.1. Further research is necessary to identify how significant a change this would be or what techniques the worm writer could employ to improve its propagation efficiency. It would seem that a SQL slammer-type worm would be far less effective in an IPv6 environment because of its inability to find hosts to infect and thus its inability to bring about the flooding result.

3.1.8.2.2 Current Technology Capabilities

The three mitigation techniques currently used in IPv4 are all still available in IPv6. There is not, however, broad IPv6 support in the host IDS products currently available. Additionally, the information provided by routers to aid in anomaly detection is not as extensive in IPv6 at this time.

3.1.8.3 Candidate Best Practices

Beyond establishing techniques to make local attack traceback easier, there are no best practice changes with virus and worm attacks. All the mechanisms from IPv4 (when the products support IPv6) work properly.

3.1.9 Translation, Transition, and Tunneling Mechanisms

Much thought and attention has been paid to how IPv4 networks will be transitioned to IPv6 networks. Additionally, work has already started on evaluating the security implications of the IPv4-to-IPv6 migration techniques. This section does not attempt to analyze these mechanisms in detail, but instead summarizes the security research efforts and provides observations.

Several approaches to transitioning from IPv4 to IPv6 networks exist. These approaches fall into the following categories:

- Dual stack
- Tunneling
- Translation

The existence of so many transition technologies creates a situation in which network designers need to understand the security implications of the transition technologies and select the appropriate transition technology for their network. The previous sections of this document assumed that the end hosts and networking infrastructure were dual stacked when discussing IPv6 native access. The following outlines some of the issues when the end hosts are not dual stacked and must rely on tunneling or translation technologies for IPv4 communications.

3.1.9.1 Issues and Observations

- With regard to IPv6 tunneling technologies and firewalls, if the network designer does not consider IPv6 tunneling when defining security policy, unauthorized traffic could possibly traverse the firewall in tunnels. This is similar to the issue with Instant Messaging (IM) and file sharing applications using TCP port 80 out of organizations with IPv4.
- As noted in many of the transition studies done, automatic tunneling mechanisms are susceptible to packet forgery and DoS attacks. These risks are the same as in IPv4, but increase the number of paths of exploitation for adversaries.

- Tunneling overlays are considered nonbroadcast multiaccess (NBMA) networks to IPv6 and require the network designer to consider this fact in the network security design. The network designer must consider this when deploying automatic or static tunneling.
- Relay translation technologies introduce automatic tunneling with third parties and additional DoS vectors. These risks do not change from IPv4, but do provide new avenues for exploitation [33]. These avenues can be limited by restricting the routing advertisements of relays to internal or external customers.
- Static IPv6 in IPv4 tunneling is preferred because explicit allows and disallows are in the policy on the edge devices.
- Translation techniques outlined for IPv6 have been analyzed [34] and shown to suffer from similar spoofing and DoS issues as IPv4-only translation technologies.
- IPv6-to-IPv4 translation and relay techniques can defeat active defense traceback efforts hiding the origin of an attack.

When focusing on host security on a dual-stack device, be aware that applications can be subject to attack on both IPv6 and IPv4. Therefore, any host controls (firewalls, VPN clients, IDSs, and so on) should block traffic from both IP versions when a block is necessary. For example, when split tunneling is disabled on an IPv4 VPN client, that VPN client should block IPv6 split tunneling as well, even if the VPN service does not expressly support IPv6. IPv4 to IPv6 transition attack tools are already available that can spoof, redirect, and launch DoS attacks.

3.1.9.2 Candidate Best Practices

General recommendations for networks when considering IPv6-to-IPv4 transition techniques include the following:

- *Use dual stack as your preferred IPv6 migration choice*–Use either native IPv4 or IPv6 access to services but not translation because the security issues are better understood and policy implementations can be simplified.
- *Use static tunneling rather than dynamic tunneling*–This allows the administrator to establish a trust relationship between tunnel endpoints and continue to implement inbound and outbound security policy.
- *Implement outbound filtering on firewall devices to allow only authorized tunneling endpoints*–Examples are filtering outbound IP Protocol 41 for 6to4 tunneling and UDP port 3544 for Teredo-based tunneling.

3.2 Attacks with Strong IPv4 and IPv6 Similarities

This section outlines attacks that are not fundamentally altered by IPv6:

- Sniffing
- Application layer attacks
- Rogue devices
- Man-in-the-middle attacks
- Flooding

3.2.1 Sniffing

Sniffing refers to the class of attacks that involves capturing data in transit across a network. The most common example of this is Tcpdump, which is included in most UNIX-like operating systems. An

adversary executing sniffing attacks can often determine login credentials or view sensitive information in plaintext protocols.

Although IPv6 provides fundamental technology to prevent sniffing with IPsec, it does not provide any simplification for the key management issues that have proved to be challenging. Until the key management issues (among others) are resolved, deployment of IPsec will be stalled and sniffing attacks will continue to be possible.

3.2.2 Application Layer Attacks

Application layer attacks refer to all the attacks performed at Layer 7 of the OSI model. This is the bulk of all attacks on the Internet today, and the vulnerabilities that enable these attacks represent the source of most of the insecurities in today's networks. General attacks such as buffer overflows, Web application attacks (Common Gateway Interface [CGI] and so on), and viruses and worms all fall into this category. IPv4 and IPv6 are both, for the most part, neutral parties to application layer attacks. Certainly if the protocol had adopted more stringent authentication of IP addresses some of these attacks could be more easily traced, but the bulk of any blame in application layer attacks lies in the affected application, not the underlying transport.

Even assuming the worldwide implementation of IPsec, application layer attacks change very little with IPv6 adoption. Even though a given connection can be cryptographically protected, there is nothing to stop an application layer attack from traversing the encrypted link and causing the same damage as if it were in the clear. The only difference is that tracing back the attack may prove easier because of the authentication in cases where Layer 3 information could otherwise be spoofed.

However, if IPsec is more ubiquitously deployed from end station to end station, without some mechanism for key, all security protections will fall to the host. Because all a firewall or IDS sees is encrypted traffic, it cannot make any decisions based on such data.

3.2.3 Rogue Devices

Rogue devices are devices introduced into the network that are not authorized. Although this could most easily be a simple unauthorized laptop, more interesting for an adversary would be a rogue wireless access point, DHCP or DNS server, router, or switch. These attacks are fairly common in IPv4 networks and are not substantially changed in IPv6. If IPsec were ever used in a more comprehensive way in the IPv6 protocol (including device bootstrap), authentication for devices could mitigate this attack somewhat. The 802.1x standard also has the potential to help here, though an undetected rogue device could funnel 802.1x authentication sequences to a compromised node acting as a AAA server while capturing valid credentials.

3.2.4 Man-in-the-Middle Attacks

Because the IPv4 and IPv6 headers have no security mechanisms themselves, each protocol relies on the IPsec protocol suite for security. In this fashion IPv6 falls prey to the same security risks posed by a man in the middle attacking the IPsec protocol suite, specifically IKE. Tools that can attack an IKE aggressive mode negotiation and derive a preshared key are documented. With this in mind, we recommend using IKE main mode negotiations when requiring the use of preshared keys. IKEv2 is expected to address this issue in the future.

3.2.5 Flooding

Though certainly the increase in IP addresses that can be spoofed may make flooding attacks more difficult to trace, the core principles of a flooding attack remain the same in IPv6. Whether a local or a

distributed DoS attack, flooding a network device or host with more traffic than it is able to process–or more than the link can transmit–is an easy way to take a resource out of service. The same techniques used to locate and trace back DoS attacks in IPv4 can be used in IPv6, though new techniques may be available. This is an area for further research, and is highlighted in Appendix A.

4 Overview of IPv6 Topology and Best-Practice Security Rules

After migrating the typical IPv4 Internet edge network to a dual-stack IPv4 or IPv6 design, the topology itself does not change, as shown in Figure 2.

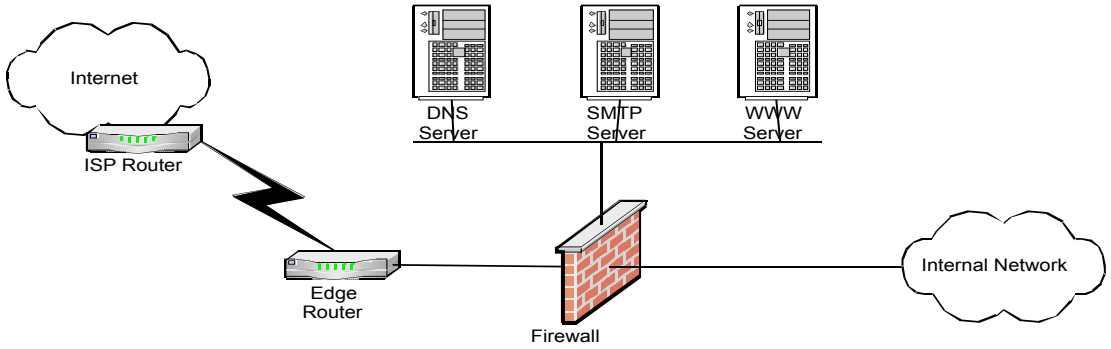

Figure 2 Candidate Design of Dual-Stack IPv4 or IPv6 Internet Edge Network

As was discussed in section 3, many threats that have different attack vectors or different levels of impact in an IPv6 network. The impact of these changes can be seen in the configurations provided in Appendix B. This section briefly highlights high-level best-practice differences in the dual-stack design vs. IPv4 only. The candidate best practices presented throughout section 3 are then included for easy reference.

Some of the main considerations for deployment of a dual-stack Internet edge are ensuring that you have good configuration change control and monitoring for your firewall and edge router. For example, the configuration of the IPv4-only firewall in the test lab was just over 200 lines long. When v6 is added, the configuration is over 300 lines long. Just like in any device, as the configuration size increases, so does the chance for error. Combine that with the fact that these hosts now have two distinct protocols on which they can be attacked as well as a lack of broad IPv6 support in current security technologies, and the chance that the adversary will find a new way into your network with IPv6 increases.

One interesting best-practice shift not already discussed in this document is how bogon filtering changes with IPv6. In IPv4, because so much of the IPv4 range has been allocated, it is generally easier to block bogons than it is to permit nonbogons. In IPv6, only three top-level aggregation identifiers (TLAs) have been allocated thus far. The Internet Assigned Numbers Authority (IANA) maintains a document [35] listing the most current list of IPv6 TLA assignments. The three currently allocated follow:

- 2001:/16 – Main IPv6 production block with sub-TLA assignments to the RIRs–Periodically tracking the /23s allocated to the regional Internet registries (RIRs) will allow even tighter protection.
- 2002:/16 – 6to4 tunneling–Hosts using 6to4 may still need to communicate with your IPv6 hosts. The IPv4 bogon list is applicable to the subsequent 32 bits.
- 3FFE:/16 – 6Bone testing–This range is deprecated and scheduled to be vacated by June 6, 2006.

Therefore, ACLs can permit these ranges (and certain multicast ranges if used) and block all other IPv6 traffic. This certainly does not prevent you from receiving spoofed traffic because the ranges that have

been allocated are immense, but it stops obviously malicious or malformed traffic using unallocated addresses.

The other main change to the filtering practices in the IPv6 portion of the dual-stack configuration is the additional ICMP types that may be necessary, as discussed in section 3.1.2.

The following lists the candidate best practices presented throughout section 3. They are listed as candidate because without more broad testing and input from the community they cannot be construed as anything more than a best guess:

- *Implement privacy extensions carefully*—Although privacy extensions are a benefit from an obscurity standpoint regarding scanning attacks, they can also make it difficult to trace problems and troubleshoot issues on a network. If a network has a misbehaving host and that host's address changes regularly, it could be quite difficult to trace the exact host or to determine if the problems are from one host or many. Better options are to use static addresses for internal communication that are MAC address-based and pseudorandom addresses for traffic destined for the Internet. In addition, this makes current audit capabilities to track worms more challenging because when we track an infection back to a particular subnet, the privacy extensions rotation of the addresses or a machine reboot could make it difficult to identify the infected end host.
- *Filter internal-use IPv6 addresses at organization border routers*—Administrators can define site-local addresses for their organization, including specific multicast addresses such as the all-routers address FF05::2. These site-local addresses can potentially lead to new avenues of attack, so administrators must filter these addresses at an organization's border routers.
- *Use standard, but nonobvious static addresses for critical systems*—Instead of standardizing on host addresses such as ::10 or ::20, try something that is more difficult for adversaries to guess, such as ::DEF1 for default gateways. This is certainly a "security through obscurity" technique, but because it involves little additional effort on the administrator's part, its use has no drawbacks. The goal here is to make it difficult for the adversary to guess the global addresses of key systems. Standardizing on a short, fixed pattern for interfaces that should not be directly accessed from the outside allows for a short filter list at the border routers.
- *Filter unneeded services at the firewall*—Like in IPv4, your public and internal systems should not be reachable on services that they do not need to be reached on. Though some are hoping that tools such as IPsec will eliminate the need for firewalls, they will be around for years to come as Layer 3 and 4 filtering is well understood. Until some nontechnical issues (such as the international politics of who controls any trust roots) are resolved, wide-scale deployment of IPsec will be impractical for both IPv4 and IPv6.
- *Selectively filter ICMP*—Because neighbor discovery uses ICMP and fragmentation is done only on end stations (which requires path maximum-transmission-unit discovery [PMTUD]), it is imperative that some ICMP messages be permitted in IPv6. That said, nonessential ICMP messages can be filtered at a firewall, as can ICMP echo and echo-reply messages, if that aspect of manageability can be sacrificed. The author's recommend that, particularly for IPv6, ICMP echo be enabled in all directions for all hosts, except that inbound ICMP echoes from the Internet to the internal network should be denied. Additionally, IPv6 requires ICMPv6 neighbor discovery-neighbor solicitation (ND-NS) and neighbor discovery-neighbor advertisement (ND-NA) messages to function (described in section 3.1.2), as well as router-solicitation (RS) and router-advertisement (RA) messages if autoconfiguration is used and RA messages are sent from the router for prefix lifetime advertisements. Finally, as in IPv4, packet-too-big messages should be broadly permitted to ensure proper functioning of PMTUD. Section 3.1.2.2.1.3 describes the ICMP messages required in more detail.
- *Maintain host and application security*—Although timely patching and host lockdown are critical elements in IPv4, they are even more critical during the early stages of IPv6 because many host

protections (firewalls, IDSs, and so on) do not yet broadly support IPv6. Additionally, it is highly likely (though testing is necessary; refer to Appendix A) that the initial introduction of IPv6 into networks will result in some hosts not being properly secured. It is necessary to focus on maintaining host security to ensure that hosts that are compromised will not become stepping stones to compromise other end hosts.

- *Determine what extension headers will be allowed through the access control device*–Network designers should match their IPv6 policy to their IPv4 IP options policy. If any IPv4 IP options are denied on the access control device, the IPv6 access control device should implement the same policies. Additionally, administrators should understand the behavior of the end-host operating system when dealing with the extension headers and dictate security policy based on that behavior. For instance, as noted earlier, the administrator should validate that end-host operating systems do not forward packets that contain a routing header.
- *Determine which ICMPv6 messages are required*–It is recommended that administrators match their policy map closely to the equivalent ICMPv4 policy with the following additions:
  - ICMPv6 Type 2 - Packet too big
  - ICMPv6 Type 4 – Parameter problem
  - ICMPv6 Type 130-132 – Multicast listener
  - ICMPv6 Type 133/134 – Router solicitation and router advertisement
  - ICMPv6 Type 135/136 – Neighbor solicitation and neighbor advertisement
- *Deny IPv6 fragments destined to an internetworking device when possible*–This will limit certain attacks against the device. However, this filtering should be tested before deployment to ensure that it does not cause problems in your particular network environment.
- *Ensure adequate IPv6 fragmentation filtering capabilities*–The combination of multiple extension headers and fragmentation in IPv6 creates the potential that the Layer 4 protocol will not be included in the first packet of a fragment set. Security monitoring devices that expect to find the Layer 4 protocol need to account for this possibility and reassemble fragments.
- *Drop all fragments with less than 1280 octets (except the last one)*–RFC 2460 section 5 says "IPv6 minimum MTU is 1280 octets." For this reason security devices may be able to drop any IPv6 fragment with less than 1280 octets unless it is the last fragment in the packet. More testing is necessary in this area, as specified in section 3.1.3.2.1. A case that should be noted is for Layer 2 firewalls and IPv4 routers transporting a tunnel. There is no requirement that IPv6 packets be 1280 octets or more between Layer 3 interfaces, just that if the packet is fragmented, the fragments must be reassembled at the receiving interface before forwarding. This is done specifically to allow tunneling over IPv4 networks where the MTU might be less than 1280. In that case, IPv4 is architecturally Layer 2.
- *Implement RFC 2827-like filtering and encourage your ISP to do the same*–At least containing spoofed traffic to the host portion of the IPv6 address provides a large benefit for at least tracing the attack back to the originating network segment.
- *Document procedures for last-hop traceback*–With the large range of spoofable addresses in a IPv6 subnet, it is critical that when an attack does occur you have mechanisms to determine the true physical source of the traffic. This generally entails some combination of Layer 2 and Layer 3 information gleaned from switches and routers.
- *Use cryptographic protections where critical*–If an application uses strong cryptographic protections, a successful spoof attack is meaningless without also subverting the cryptographic functions on the device.

- *Use static neighbor entries for critical systems*—In highly sensitive environments you can specify that a system has a static entry to its default router and avoid many of the typical neighbor-discovery attacks. This is a very administratively burdensome practice and should not be undertaken lightly.
- *Implement ingress filtering of packets with IPv6 multicast source addresses*—There is no valid reason for a multicast source address, so the administrator should drop any packets with a multicast source address at the border of the network.
- *Use traditional authentication mechanisms on BGP and IS-IS.*
- *Use IPsec to secure protocols such as OSPFv3 and RIPng*—This is dependant on functioning vendor implementations.
- *Use IPv6 hop limits to protect network devices*—Investigate vendor implementations of IPv6 hop limits to protect the protocol stack from attack. For instance, a basic technique is to start the time to live (TTL) of 255 for a valid peer and ensure that the resulting TTL accepted by the router is high enough to prevent acceptance of a spoofed packet that has come from a different part of the infrastructure.
- *Use dual stack as your preferred IPv6 migration choice*—Use either native IPv4 or IPv6 access to services but not translation because the security issues are better understood and policy implementations can be simplified.
- *Use static tunneling rather than dynamic tunneling*—This allows the administrator to establish a trust relationship between tunnel endpoints and continue to implement inbound and outbound security policy.
- *Implement outbound filtering on firewall devices to allow only authorized tunneling endpoints*—Examples are filtering outbound IP Protocol 41 for 6to4 tunneling and UDP port 3544 for Teredo-based tunneling.

5 Summary

As shown in this paper, IPv6 has both benefits and drawbacks from a security standpoint. The opportunity to ensure secure IPv6 deployments from the outset rather than a slow migration toward security, as occurred with IPv4, should be strongly considered by the Internet community. However, the amount of attention that IPv6 security has so far received is quite low, and new considerations will certainly be uncovered. Without adequate training and attention on the part of network operators to the new considerations with IPv6 security, it will be very difficult to ensure a smooth transition (or any transition at all) to IPv6.

This paper has introduced you to the security issues and candidate best practices surrounding the introduction of IPv6 into a network, with or without IPsec. With this understanding, you should be able to identify areas that need further research in your own network, or begin to think about how a migration to dual-stack IPv6 might occur. When ready, the information in this paper will aid you in modifying your security policies to account for IPv6 and in the beginning to test the suitability of the best practices in this document as applied to your own network. Ongoing research will continue to refine the contents of this paper and provide more information on candidate best practices and technology updates.

6 Acknowledgments

The authors wish to thank all the reviewers of this document, including Steve Acheson, Bora Akyol, John Bashinski, Liang Gao, Craig Huegen, Franjo Majstor, Shannon McFarland, Steve Pollock, Gavin Reid, Mike Schiffman, and Andrew Wright, and a special thanks to Tony Hain for his extensive support of this document.

7 Change Log

11 MAR 2004 – Initial Version 1.0 Released

8 References

[1] S Deering, R Hinden, “Internet Protocol, Version 6 (IPv6) Specification” (December 1998), RFC 2460 at http://www.ietf.org/rfc/rfc2460.txt

[2] J Postel, “Internet Protocol, DARPA Internet Program Protocol Specification” (September 1981), RFC 0791 at http://www.ietf.org/rfc/rfc0791.txt

[3] S Kent, R Atkinson, “Security Architecture for the Internet Protocol” (November 1998), RFC 2401 at http://www.ietf.org/rfc/rfc2401.txt

[4] The Internet Engineering Task Force (IETF) at http://www.ietf.org

[5] Press Release, “Defense Department Will Require IPv6 Compliance, Says DoD's John Osterholz” (June 26, 2003), at http://biz.yahoo.com/iw/030626/054991.html

[6] D Johnson, C Perkins, J Arkko, “Mobility Support in IPv6 draft-ietf-mobileip-ipv6-24.txt” (June 30, 2003), at http://www.ietf.org/internet-drafts/draft-ietf-mobileip-ipv6-24.txt

[7] Nmap at http://www.insecure.org/nmap/

[8] T Narten, R Draves, “Privacy Extensions for Stateless Address Autoconfiguration in IPv6” (January 2001), RFC 3041 at http://www.ietf.org/rfc/rfc3041.txt

[9] R Hinden, S Deering, “IPv6 Multicast Address Assignments” (July 1998), RFC 2375 at http://www.ietf.org/rfc/rfc2375.txt

[10] P Savola, “Security of IPv6 Routing Header and Home Address Options” (December 2002), at http://www.6net.org/publications/standards/draft-savola-ipv6-rh-ha-security-03.txt

[11] R Hinden, S Deering, “IP Version 6 Addressing Architecture” (July 1998), RFC 2373 at http://www.ietf.org/rfc/rfc2373.txt

[12] S Woolf, NANOG, “’Anycasting’ f.root-servers.net” (January 2003), at http://www.nanog.org/mtg-0302/ppt/suzanne.pdf

[13] L Dondeti, T Hardjono, B Haberman, “Anycast security requirements” at http://www.securemulticast.org/smug12-Dondeti.PDF

[14] M Baugher, B Weis, T Harjono, H Harney, “Group Domain of Interpretation” (July 2003), RFC 3547 at http://www.ietf.org/rfc/rfc3547.txt

[15] C Shannon, D Moore, K Claffy, “Characteristics of Fragmented IP Traffic on Internet Links” (2001), at http://www.caida.org/outreach/papers/2001/Frag/frag.pdf

[16] P Ferguson, D Senie, “Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing” (May 2000), RFC 2827, BCP38 at http://www.ietf.org/rfc/rfc2827.txt

[17] IANA, “Internet Protocol v4 Address Space” at http://www.iana.org/assignments/ipv4-address-space.

[18] IANA, “Special-Use IPv4 Addresses” (September 2002), RFC 3330 at http://www.ietf.org/rfc/rfc3330.txt.

[19] IETF ITRACE Working Group at http://www.ietf.org/html.charters/OLD/itrace-charter.html

[20] Cisco Documentation “Configuring DHCP Snooping” at http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/12_1_13/config/dhcp.htm

[21] Cisco Documentation, Cisco Catalyst 4500, “Configuring Dynamic ARP Inspection” at http://www.cisco.com/en/US/products/hw/switches/ps4324/products_configuration_guide_chapter09186a008019d0ca.html

[22] IETF SEND Working Group at http://www.ietf.org/html.charters/send-charter.html

[23] D Senie, “Changing the Default for Directed Broadcasts in Routers” (August 1999), RFC 2644 at http://www.ietf.org/rfc/rfc2644.txt

[24] A Conta, S Deering, “Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification” (December 1998), RFC 2463 at http://www.ietf.org/rfc/rfc2463.txt

[25] P Marques, F DuPont, “Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing” (March 1999), RFC 2545 at http://www.ietf.org/rfc/rfc2545.txt

[26] R Callon, “Use of OSI IS-IS for routing in TCP/IP and dual environments” (December 1990), RFC 1195 at http://www.ietf.org/rfc/rfc1195.txt

[27] C Hopps, “Routing IPv6 with IS-IS” (January 2003), at http://www.ietf.org/internet-drafts/draft-ietf-isis-ipv6-05.txt

[28] T Li, R Atkinson, “Intermediate System to Intermediate System (IS-IS) Cryptographic Authentication” (July 2003), RFC 3567 at http://www.ietf.org/rfc/rfc3567.txt

[29] R Coulton, D Ferguson, J Moy, “OSPF for IPv6” (December 1999), RFC 2740 at http://www.ietf.org/rfc/rfc2740.txt

[30] G Malkin, “RIPng Protocol Applicability Statement” (January 1997), RFC 2081 at http://www.ietf.org/rfc/rfc2081.txt

[31] V Gill, J Heasley, D Meyer, “The Generalized TTL Security Mechanism” (October 2003) at http://www.ietf.org/internet-drafts/draft-gill-gtsh-04.txt

[32] D Moore et. al., “Inside the Slammer Worm” at http://www.caida.org/outreach/papers/2003/sapphire2/

[33] P Savola, “Security Considerations for 6to4” (October 2003), at http://www.ietf.org/internet-drafts/draft-ietf-v6ops-6to4-security-00.txt

[34] Okazaki, A Desai, “NAT-PT Security Considerations” (June 2003), at http://www.ietf.org/internet-drafts/draft-okazaki-v6ops-natpt-security-00.txt

[35] IANA, “IPv6 Top Level Aggregation Identifier Assignments” (July 2003) at http://www.iana.org/assignments/ipv6-tla-assignments

Appendix A: Current and Future Directions for Research

Numerous areas regarding IPv6 need further research. This section briefly outlines some ideas.

First is the notion of system identification within an organization. With the advent of privacy extensions and the size of the IPv6 ranges in use, identifying systems within an organization and, in particular, identifying misbehaving hosts can be quite difficult and problematic. This document explores these problems and some potential solutions, but more research is needed in this area.

Second, the increased dependence on multicast addresses in IPv6 could have some interesting implications with flooding attacks. For example, all routers, NTP servers, and so on have site-specific multicast addresses. Can these addresses be used as a form of amplification attack much like the smurf attack in IPv4?

Third, because neighbor discovery is a new addition to IPv6 and because it is an essential component of a well-run IPv6 network, it should be exhaustively tested from a security standpoint. For example, can the neighbor-discovery cache fall victim to a resource starvation attack in any of the currently deployed neighbor-discovery implementations? Can the CPU of a device be exhausted by processing neighbor-discovery information?

Fourth, with a new header configuration, new extension headers, and new ICMP message types, there may be novel ways to deal with flooding attacks. At the very least, the various extensions proposed for IPv4 to deal with flooding attacks should be examined for applicability in IPv6.

Fifth, because IPv6 is new and security information on the protocol is not widespread, it is the opinions of the authors that a large number of dual-stack hosts may be more exposed to attack with IPv6 than they currently are in IPv4. Within the limits of the law, it would be very useful to actively scan such systems to confirm or debunk this theory and get a sense of the magnitude of the issues with currently deployed IPv6 security. This same testing can include vulnerability scans of default operating system installs against IPv4 and IPv6.

Finally, though some work has been done in this area, it would be useful to examine how some of the recent Internet worms might have fared in an IPv6 environment. This should certainly examine worms such as SQL slammer and also theoretical worms that have not yet seen release. New ways that future worms might better react to with respect to the size of the IPv6 range should also be explored. Such research should consider if any differences in propagation rate occur with only our current TLAs allocated or some time in the far future with higher-speed links and a much larger percentage of addresses allocated from the 128-bit IPv6 range. The expected increase in heterogeneity of IPv6 devices as compared to IPv4 should also be considered (refrigerators, and so on being connected to the network).

Appendix B: Configurations from the Lab

Figure B1 shows the lab topology first shown just for the IPv4 component of the lab:

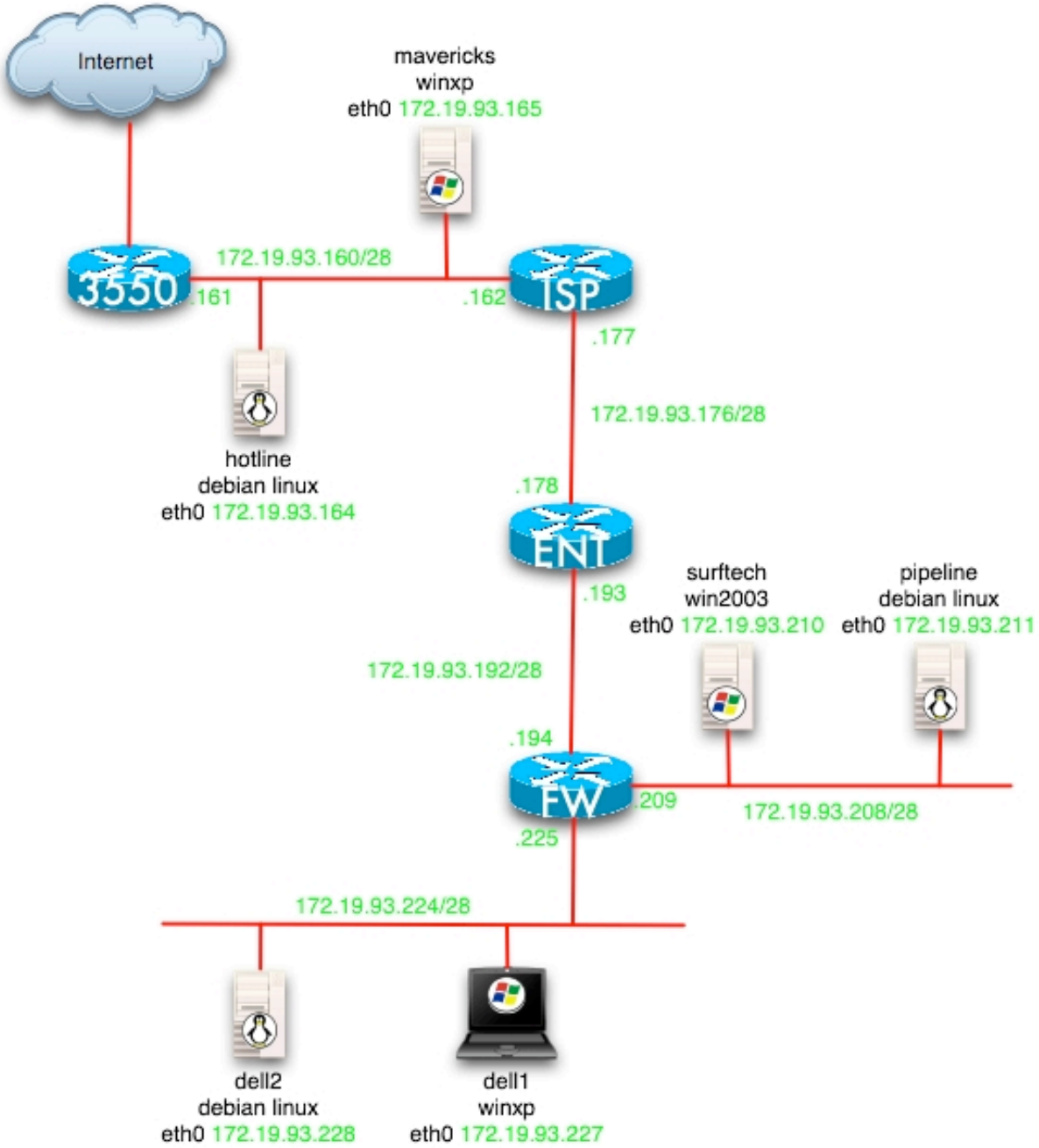

Figure B1 Lab Topology – IPv4

The organization represented by Figure B1 controls the configuration of router "FW" and router "ENT". ENT is acting as the WAN router to the ISP, and FW is acting as the FW router between the Internet and the internal network. The relevant configuration components of router ENT are as follows:

```
!
version 12.3
!
hostname ENT
!
boot-start-marker
boot system flash:c2600-ik9o3s3-mz.123-3.bin
boot-end-marker
ip cef
!
!
interface FastEthernet0/0
 ip address 172.19.93.178 255.255.255.240
! This is the primary inbound ACL for basic bogon filtering
! and anti-spoof filtering
 ip access-group 101 in
! Enable unicast RPF checking for anti-spoofing
 ip verify unicast reverse-path
! Basic best practices regarding the handling of certain ICMP
! types
 no ip redirects
 no ip unreachables
 no ip proxy-arp
!
interface FastEthernet0/1
 ip address 172.19.93.193 255.255.255.240
! This is the last point of egress filtering before traffic
! is sent to the ISP
 ip access-group 102 in
! Enable unicast RPF checking for anti-spoofing
 ip verify unicast reverse-path
! Basic best practices regarding the handling of certain ICMP
! types
 no ip redirects
 no ip unreachables
 no ip proxy-arp
!
! This entire lab was built using static routing
ip route 0.0.0.0 0.0.0.0 172.19.93.177
ip route 172.19.93.208 255.255.255.240 172.19.93.194
ip route 172.19.93.224 255.255.255.240 172.19.93.194
!
!
```



```
logging trap warnings
! Log back to the syslog daemon on dell2
logging 172.19.93.228
! ACL limiting access to NTP to the actual NTP server we are
! communicating with
access-list 96 permit 171.68.10.80
access-list 96 deny   any log
! Bogon filtering (RFC 3330) Note that RFC 1918 private
! ranges are not filtered since these are used in the test
! lab. Also RFC 2827 filtering is implemented via unicast RPF
! filtering as opposed to explicit ACL entries
access-list 101 deny   ip 0.0.0.0 0.255.255.255 any
access-list 101 deny   ip 127.0.0.0 0.255.255.255 any
access-list 101 deny   ip 169.254.0.0 0.0.255.255 any
access-list 101 deny   ip 192.0.2.0 0.0.0.255 any
access-list 101 deny   ip 198.18.0.0 0.1.255.255 any
! Filter multicast ranges, not a good idea if you route
! multicast with your ISP
access-list 101 deny   ip 224.0.0.0 15.255.255.255 any
access-list 101 deny   ip 240.0.0.0 15.255.255.255 any
! Standard IPv4 ICMP filtering best practices, drop fragments
! and allow messages which are needed. Note, in this lab ICMP
! echo and echo-reply was broadly permitted to ease testing
! in a production network more restrictive filtering is
! probably warranted.
access-list 101 deny   icmp any any fragments
access-list 101 permit icmp any any echo
access-list 101 permit icmp any any echo-reply
access-list 101 permit icmp any any packet-too-big
access-list 101 permit icmp any any time-exceeded
access-list 101 deny   icmp any any
access-list 101 permit ip any any
! Inbound on the internal facing interface basic ICMP
! filtering is all that is present.
! RFC 2827 filtering is implemented using unicast RPF
! filtering
access-list 102 deny   icmp any any fragments
access-list 102 permit icmp any any echo
access-list 102 permit icmp any any echo-reply
access-list 102 permit icmp any any packet-too-big
access-list 102 permit icmp any any time-exceeded
access-list 102 deny   icmp any any
access-list 102 permit ip any any
```



```
!
ntp clock-period 17179981
ntp access-group peer 96
ntp server 171.68.10.80
!
end
```

Note that not all router hardening steps are shown here nor should this be assumed a fully locked down IPv4 router config. The following is the IPv4 configuration for the FW device:

```
!
version 12.3
!
hostname FW
!
boot-start-marker
boot system tftp c2600-bino3s-mz.ipv6_eft 172.19.93.228
boot-end-marker
!
ip cef
!
!
! The following commands setup the parameters for the IOS
! firewall, they have not been tuned in any way. The
! inspection name for the v4 firewall is "v4_fw".
ip inspect audit-trail
ip inspect max-incomplete low 150
ip inspect max-incomplete high 250
ip inspect one-minute low 100
ip inspect one-minute high 200
ip inspect udp idle-time 20
ip inspect dns-timeout 3
ip inspect tcp idle-time 1800
ip inspect tcp finwait-time 3
ip inspect tcp synwait-time 15
ip inspect tcp max-incomplete host 40 block-time 0
ip inspect name v4_fw tcp timeout 300
ip inspect name v4_fw udp
ip inspect name v4_fw tftp
ip inspect name v4_fw http
ip inspect name v4_fw fragment maximum 256 timeout 1
!
!
interface FastEthernet0/0
 ip address 172.19.93.194 255.255.255.240
```



```
! This is the primary inbound ACL for traffic sent from the
! Internet which has passed by the ENT router. As a note,
! with IOS FW, the return traffic does not need to be
! explicitly permitted. This functionality is facilitated
! with the “ip inspect” command.
 ip access-group 101 in
! Enable unicast RPF checking for anti-spoofing
 ip verify unicast reverse-path
! Basic best practices regarding the handling of certain ICMP
! types
 no ip redirects
 no ip unreachables
 no ip proxy-arp
! enable IOS FW functionality on this interface
 ip inspect v4_fw in
!
interface FastEthernet0/1
 ip address 172.19.93.225 255.255.255.240
! This is the primary ACL for traffic originated from the
! internal network
 ip access-group 103 in
! Enable unicast RPF checking for anti-spoofing
 ip verify unicast reverse-path
! Basic best practices regarding the handling of certain ICMP
! types
 no ip redirects
 no ip unreachables
 no ip proxy-arp
! enable IOS FW functionality on this interface
 ip inspect v4_fw in
!
interface Ethernet1/0
 ip address 172.19.93.209 255.255.255.240
! This is the primary ACL for traffic originated within the
! public server segment (DMZ)
 ip access-group 102 in
! Enable unicast RPF checking for anti-spoofing
 ip verify unicast reverse-path
! Basic best practices regarding the handling of certain ICMP
! types
 no ip redirects
 no ip unreachables
 no ip proxy-arp
```



```
! enable IOS FW functionality on this interface
 ip inspect v4_fw in
!
!
no ip http server
no ip http secure-server
ip classless
ip route 0.0.0.0 0.0.0.0 172.19.93.193
!
!
! Enable logging to the syslog daemon on dell2
logging 172.19.93.228
! ACL to control access to NTP
access-list 96 permit 171.68.10.80
access-list 96 deny   any log
! Access-list 101 is the inbound ACL for outside traffic
! ICMP filtering best practices
access-list 101 deny   icmp any any fragments
access-list 101 permit icmp any any echo
access-list 101 permit icmp any any echo-reply
access-list 101 permit icmp any any packet-too-big
access-list 101 permit icmp any any time-exceeded
access-list 101 deny   icmp any any
! ACL entry to facilitate remote lab access (not needed in
! production)
access-list 101 permit ip any host 172.19.93.194
! ACL entry allowing SSH access to the entire DMZ network for
! testing, Not needed or desirable in production
access-list 101 permit tcp any 172.19.93.208 0.0.0.15 eq 22
! Standard ACL entries to permit access to the SMTP and DNS
! listeners at this IP address, these functions are merged on
! a single device in the test lab but in most situations will
! be on separate machines
access-list 101 permit tcp any host 172.19.93.211 eq smtp
access-list 101 permit tcp any host 172.19.93.211 eq domain
access-list 101 permit udp any host 172.19.93.211 eq domain
! Permit access to the .210 on web and ftp
access-list 101 permit tcp any host 172.19.93.210 eq www
access-list 101 permit tcp any host 172.19.93.210 eq ftp
! ACL entry to permit remote SSH access to the dell2 machine,
! not needed or desirable in a production network
access-list 101 permit tcp any host 172.19.93.228 eq 22
! ACL entries to permit syslog and TFTP access from the ENT
```

```
! router to the syslod and TFTP listeners on dell2
access-list 101 permit udp host 172.19.93.193 host 172.19.93.228 eq syslog
access-list 101 permit udp host 172.19.93.193 host 172.19.93.228 eq tftp
! deny all other traffic and log the event including the
! input source
access-list 101 deny   ip any any log-input
! Standard ICMP filtering
access-list 102 deny   icmp any any fragments
access-list 102 permit icmp any any echo
access-list 102 permit icmp any any echo-reply
access-list 102 permit icmp any any packet-too-big
access-list 102 permit icmp any any time-exceeded
access-list 102 deny   icmp any any
! Permit both public servers to send syslog to the syslog
! listener on dell2
access-list 102 permit udp 172.19.93.210 0.0.0.1 host 172.19.93.228 eq syslog
! Permit SSH and TFTP from these servers to dell2 for testing
access-list 102 permit udp 172.19.93.210 0.0.0.1 host 172.19.93.228 eq tftp
access-list 102 permit tcp 172.19.93.210 0.0.0.1 host 172.19.93.228 eq 22
! Permit the outside SMTP server to transfer incoming mail
! via SMTP
access-list 102 permit tcp host 172.19.93.211 host 172.19.93.228 eq smtp
! Deny any other access to the internal network
access-list 102 deny   ip any 172.19.93.224 0.0.0.15
! Allow the two public servers to initiate outbound requests
! for DNS, SSH, WWW Proxy, and VNC. In a production network
! this would be far more limited but these permits were
! entered for testing.
access-list 102 permit tcp 172.19.93.210 0.0.0.1 any eq domain
access-list 102 permit udp 172.19.92.210 0.0.0.1 any eq domain
access-list 102 permit tcp 172.19.93.210 0.0.0.1 any eq 22
access-list 102 permit tcp 172.19.93.210 0.0.0.1 any eq www
access-list 102 permit tcp 172.19.93.210 0.0.0.1 any eq 3128
access-list 102 permit tcp 172.19.93.210 0.0.0.1 any eq 1080
! Deny any other traffic and log the input source
access-list 102 deny   ip any any log-input
! Basic ICMP filtering
access-list 103 deny   icmp any any fragments
access-list 103 permit icmp any any echo
access-list 103 permit icmp any any echo-reply
access-list 103 permit icmp any any packet-too-big
access-list 103 permit icmp any any time-exceeded
access-list 103 deny   icmp any any
```



```
! Permit SSH access from the internal network to the public
! servers. In a production network this should be locked down
! to the specific hosts that need access.
access-list 103 permit tcp 172.19.93.224 0.0.0.15 172.19.93.210 0.0.0.1 eq 22
! Permit web and ftp requests of surftech
access-list 103 permit tcp 172.19.93.224 0.0.0.15 host 172.19.93.210 eq www
access-list 103 permit tcp 172.19.93.224 0.0.0.15 host 172.19.93.210 eq ftp
! Permit mail and DNS requests of pipeline, In a production
! network this would be much more locked down to the specific
! hosts required
access-list 103 permit tcp 172.19.93.224 0.0.0.15 host 172.19.93.211 eq smtp
access-list 103 permit tcp 172.19.93.224 0.0.0.15 host 172.19.93.211 eq domain
access-list 103 permit udp 172.19.93.224 0.0.0.15 host 172.19.93.211 eq domain
! Deny any other traffic to the DMZ and log
access-list 103 deny   ip any 172.19.93.208 0.0.0.15 log-input
! Allow the internal network to initiate outbound requests
! for DNS, SSH, WWW Proxy, and VNC. In a production network
! this would be far more limited but these permits were
! entered for testing. In almost all cases additional
! services need to be opened up to the Internet to enable
! other applications in a production network.
access-list 103 permit tcp 172.19.93.224 0.0.0.15 any eq 22
access-list 103 permit tcp 172.19.93.224 0.0.0.15 any eq telnet
access-list 103 permit tcp 172.19.93.224 0.0.0.15 any eq www
access-list 103 permit tcp 172.19.93.224 0.0.0.15 any eq domain
access-list 103 permit udp 172.19.93.224 0.0.0.15 any eq domain
access-list 103 permit tcp 172.19.93.224 0.0.0.15 any eq 3128
access-list 103 permit tcp 172.19.93.224 0.0.0.15 any eq 1080
! Allow outbound TFTP, normally not needed but used for
! testing
access-list 103 permit udp 172.19.93.224 0.0.0.15 any eq tftp
! Allow TFTP responses from dell2 to the FW inside interface.
! This is needed because the IOS FW does not operate on
! traffic it originates itself.
access-list 103 permit udp host 172.19.93.228 eq tftp host 172.19.93.225 gt 1023
! Deny all other traffic and log
access-list 103 deny   ip any any log-input
!
ntp clock-period 17208089
ntp access-group peer 96
ntp server 171.68.10.80
!
end
```


The lab topology now with v6 addresses added in addition to v4 is shown in figure B2:

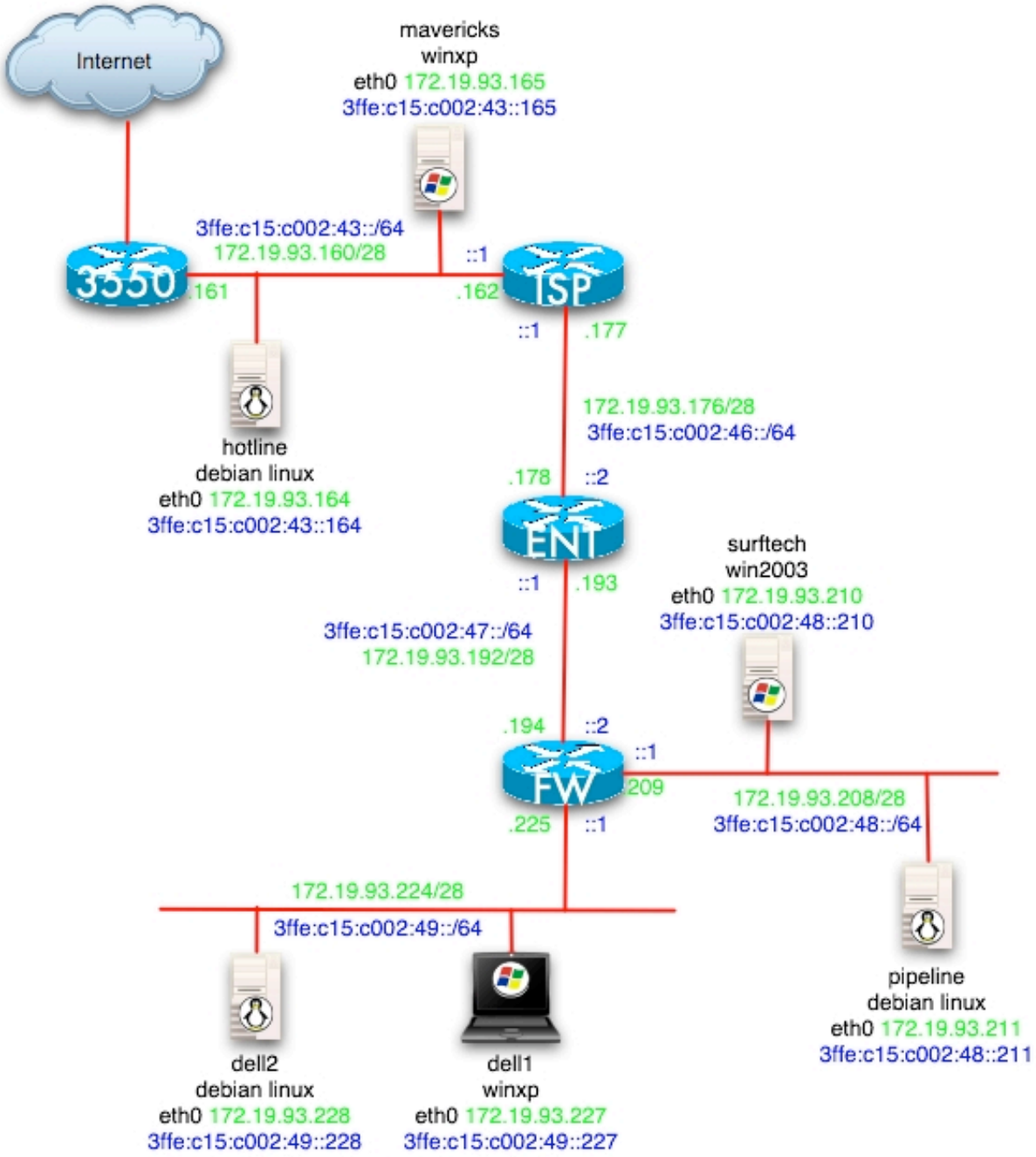

Figure B2 Lab Topology – IPv4 and IPv6

The configuration for device "ENT" is as follows showing the IPv6 additions and not the v4 elements:

```
!
version 12.3
!
hostname ENT
!
```

```
boot-start-marker
boot system flash:c2600-ik9o3s3-mz.123-3.bin
boot-end-marker
!
ipv6 unicast-routing
ipv6 cef
!
interface FastEthernet0/0
 ipv6 address 3FFE:C15:C002:46::2/64
 ipv6 traffic-filter inbound in
 ipv6 rip v6 enable
 ipv6 cef
!
interface FastEthernet0/1
 ipv6 address 3FFE:C15:C002:47::1/64
 ipv6 traffic-filter outbound in
 ipv6 rip v6 enable
 ipv6 cef
!
ipv6 router rip v6
!
!
ipv6 access-list inbound
! deny all traffic with a routing header or undetermined
! transport
 deny ipv6 any any routing
 deny ipv6 any any undetermined-transport
! permit link local traffic between routers
 permit ipv6 FE80::/10 FE80::/10
! permit rip traffic
 permit ipv6 FE80::/10 host FF02::9
! permit assigned TLAs to talk to our subnets, note that this is a
! significant departure from v4 bogon filtering in that since
! only 3 TLAs have actually been assigned, it is easier to expressly
! permit these TLAs then deny all other traffic than it is to block
! any of the special use, multicast, or other traffic normally
! associated with bogon filtering
 permit ipv6 2001::/16 host 3FFE:C15:C002:46::2
 permit ipv6 2001::/16 3FFE:C15:C002:47::/64
 permit ipv6 2001::/16 3FFE:C15:C002:48::/64
 permit ipv6 2001::/16 3FFE:C15:C002:49::/64
 permit ipv6 2002::/16 host 3FFE:C15:C002:46::2
 permit ipv6 2002::/16 3FFE:C15:C002:47::/64
```



```
 permit ipv6 2002::/16 3FFE:C15:C002:48::/64
 permit ipv6 2002::/16 3FFE:C15:C002:49::/64
 permit ipv6 3FFE::/16 host 3FFE:C15:C002:46::2
 permit ipv6 3FFE::/16 3FFE:C15:C002:47::/64
 permit ipv6 3FFE::/16 3FFE:C15:C002:48::/64
 permit ipv6 3FFE::/16 3FFE:C15:C002:49::/64
! permit ND messages
 permit icmp any any nd-na
 permit icmp any any nd-ns
! deny all other traffic
 sequence 210 deny ipv6 any any log
!
ipv6 access-list outbound
! deny all traffic with a routing header or undetermined
! transport
 deny ipv6 any any routing
 deny ipv6 any any undetermined-transport
! permit link local traffic between routers
 permit ipv6 FE80::/10 FE80::/10
! permit rip traffic
 permit ipv6 FE80::/10 host FF02::9
! permit our subnets to talk to the TLAs (and the local router)
 permit ipv6 3FFE:C15:C002:47::/64 host 3FFE:C15:C002:47::1
 permit ipv6 3FFE:C15:C002:47::/64 2001::/16
 permit ipv6 3FFE:C15:C002:47::/64 2002::/16
 permit ipv6 3FFE:C15:C002:47::/64 3FFE::/16
 permit ipv6 3FFE:C15:C002:48::/64 host 3FFE:C15:C002:47::1
 permit ipv6 3FFE:C15:C002:48::/64 2001::/16
 permit ipv6 3FFE:C15:C002:48::/64 2002::/16
 permit ipv6 3FFE:C15:C002:48::/64 3FFE::/16
 permit ipv6 3FFE:C15:C002:49::/64 host 3FFE:C15:C002:47::1
 permit ipv6 3FFE:C15:C002:49::/64 2001::/16
 permit ipv6 3FFE:C15:C002:49::/64 2002::/16
 permit ipv6 3FFE:C15:C002:49::/64 3FFE::/16
! permit ND messages
 permit icmp any any nd-na
 permit icmp any any nd-ns
! deny all other traffic
 deny ipv6 any any log
!
end
```


The configuration for the IPv6 portion of device FW is as follows. Note that bogon/TLA filtering is not done here because it has already been done on device ENT. For extra protection do the filtering in both places.

```
!
version 12.3
!
hostname FW
!
! note this is an EFT image used for this testing. IPv6 stateful firewalling is
! not currently shipping in IOS as of this writing
boot system tftp c2600-bino3s-mz.ipv6_eft 172.19.93.228
!
ipv6 unicast-routing
!
! Default values were used for the IPv6 firewalling variables
ipv6 inspect audit-trail
ipv6 inspect max-incomplete low 150
ipv6 inspect max-incomplete high 250
ipv6 inspect one-minute low 100
ipv6 inspect one-minute high 200
ipv6 inspect udp idle-time 20
ipv6 inspect tcp idle-time 1800
ipv6 inspect tcp finwait-time 3
ipv6 inspect tcp synwait-time 15
ipv6 inspect tcp max-incomplete host 40 block-time 0
ipv6 inspect name v6_fw tcp timeout 300
ipv6 inspect name v6_fw udp
ipv6 inspect name v6_fw icmp
!
interface FastEthernet0/0
 ipv6 address 3FFE:C15:C002:47::2/64
 ipv6 traffic-filter outside in
 ipv6 inspect v6_fw in
 ipv6 rip v6 enable
!
interface FastEthernet0/1
 ipv6 address 3FFE:C15:C002:49::1/64
 ipv6 traffic-filter inside in
 ipv6 inspect v6_fw in
!
interface Ethernet1/0
 ipv6 address 3FFE:C15:C002:48::1/64
 ipv6 traffic-filter dmz in
```



```
 ipv6 inspect v6_fw in
 no cdp enable
!
ipv6 router rip v6
 redistribute connected
!
! Notice that the ACLs in the v6 portion of these Acls are far more restrictive
! than the filtering done in v4. This is because the test infrastructure used
! in our lab used IPv4 which required additional permit statements. The v6
! filtering shown here is much closer to the filtering you might expect to
! actually put in place on a firewall. The notable exception is the permissive
! ICMP statements which should be locked down and the outbound access possible
! from the DMZ
!
ipv6 access-list dmz
! deny all traffic with a routing header or undetermined
! transport
 deny ipv6 any any routing
 deny ipv6 any any undetermined-transport
! ICMP filtering
 permit icmp any any echo-request
 permit icmp any any echo-reply
 permit icmp any any packet-too-big
 permit icmp any any time-exceeded
 permit icmp any any parameter-problem
 permit icmp any any nd-na
 permit icmp any any nd-ns
 deny icmp any any
! Permit SMTP transfer from pipeline to dell2
 permit tcp host 3FFE:C15:C002:48::211 host 3FFE:C15:C002:49::228 eq smtp
! deny all other traffic to the internal network
 deny ipv6 any 3FFE:C002:49::/64 log
! allow dmz systems to initiate DNS and web requests (in a production network
! this would be much more tightly locked down).
 permit tcp 3FFE:C15:C002:48::/64 any eq domain
 permit udp 3FFE:C15:C002:48::/64 any eq domain
 permit tcp 3FFE:C15:C002:48::/64 any eq www
! deny all other traffic
 deny ipv6 any any log
!
ipv6 access-list outside
! deny all traffic with a routing header or undetermined
! transport
```



```
 deny ipv6 any any routing
 deny ipv6 any any undetermined-transport
! permit link local traffic between routers
 permit ipv6 FE80::/10 FE80::/10
! permit rip traffic
 permit ipv6 FE80::/10 host FF02::9
! ICMP filtering
 permit icmp any any echo-request
 permit icmp any any echo-reply
 permit icmp any any packet-too-big
 permit icmp any any time-exceeded
 permit icmp any any parameter-problem
 permit icmp any any nd-na
 permit icmp any any nd-ns
 deny icmp any any
! permit any outside device to talk SMTP or DNS to pipeline
 permit tcp any host 3FFE:C15:C002:48::211 eq smtp
 permit tcp any host 3FFE:C15:C002:48::211 eq domain
 permit udp any host 3FFE:C15:C002:48::211 eq domain
! permit any outside device to talk WWW or FTP to surftech
 permit tcp any host 3FFE:C15:C002:48::210 eq www
 permit tcp any host 3FFE:C15:C002:48::210 eq ftp
! Deny all other traffic
 deny ipv6 any any log
!
ipv6 access-list inside
! deny all traffic with a routing header or undetermined
! transport
 deny ipv6 any any routing
 deny ipv6 any any undetermined-transport
! ICMP filtering
 permit icmp any any echo-request
 permit icmp any any echo-reply
 permit icmp any any packet-too-big
 permit icmp any any time-exceeded
 permit icmp any any parameter-problem
 permit icmp any any nd-na
 permit icmp any any nd-ns
! Note the addition of the router-soliciation permit since there are
! user stations on this subnet using address autoconfiguration
 permit icmp any any router-solicitation
 deny icmp any any
! permit the internal network to query the DMZ servers on the relevant services
```



```
 permit tcp 3FFE:C15:C002:49::/64 host 3FFE:C15:C002:48::210 eq www
 permit tcp 3FFE:C15:C002:49::/64 host 3FFE:C15:C002:48::210 eq ftp
 permit tcp 3FFE:C15:C002:49::/64 host 3FFE:C15:C002:48::211 eq smtp
 permit tcp 3FFE:C15:C002:49::/64 host 3FFE:C15:C002:48::211 eq domain
 permit udp 3FFE:C15:C002:49::/64 host 3FFE:C15:C002:48::211 eq domain
! deny all other traffic to the DMZ
 deny ipv6 any 3FFE:C15:C002:48::/64
! allow outbound DNS and W                    Pv6 Internet
 permit tcp 3FFE:C15:C002:49::/64 any eq www
 permit tcp 3FFE:C15:C002:49::/64 any eq domain
 permit udp 3FFE:C15:C002:49::/64 any eq domain
! Deny all other traffic
 deny ipv6 any any log
!
end
```