CISCO

SECURITY

# IPv6 Security

Information assurance for the next-generation
Internet Protocol

**Scott Hogg**, CCIE® No. 5133
**Eric Vyncke**

ciscopress.com

# IPv6 Security

Scott Hogg and Eric Vyncke

Copyright© 2009 Cisco Systems, Inc.

Published by:
Cisco Press
800 East 96th Street
Indianapolis, IN 46240 USA

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without written permission from the publisher, except for the inclusion of brief quotations in a review.

Printed in the United States of America

First Printing December 2008

Library of Congress Cataloging-in-Publication Data:

Hogg, Scott.

IPv6 security / Scott Hogg, Eric Vyncke.

p. cm.

Includes bibliographical references and index.

ISBN-13: 978-1-58705-594-2 (pbk.)

ISBN-10: 1-58705-594-5

1. Computer networks—Security measures. 2. TCP/IP (Computer network protocol) I. Vyncke, Eric. II. Title.

TK5105.59.H637 2009

005.8—dc22

2008047255

ISBN-13: 978-1-58705-594-2

ISBN-10: 1-58705-594-5

## Warning and Disclaimer

This book is designed to provide information about the security aspects of the IPv6 protocol. Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied.

The information is provided on an “as is” basis. The authors, Cisco Press, and Cisco Systems, Inc., shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the discs or programs that may accompany it.

The opinions expressed in this book belong to the author and are not necessarily those of Cisco Systems, Inc.

## Trademark Acknowledgments

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Cisco Press or Cisco Systems, Inc., cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

# Introduction

Internet Protocol version 6 (IPv6) is the next version of the protocol that is used for communications on the Internet. IPv6 is a protocol that has been in existence for many years, but it has not yet replaced IPv4. IPv4 has some limitations that were not anticipated when it was first created. Because IPv6 overcomes many of these limitations, it is the only viable long-term replacement for IPv4.

While the migration to IPv6 has started, it is still in its early stages. Many international organizations already have IPv6 networks, the U.S. federal organizations are working on their transitions to IPv6, and others are contemplating what IPv6 means to them. However, many organizations already have IPv6 running on their networks and they do not even realize it. Many computer operating systems now default to running both IPv4 and IPv6, which could cause security vulnerabilities if one is less secure than the other. IPv6 security vulnerabilities currently exist, and as the popularity of the IPv6 protocol increases, so do the number of threats.

When a security officer wants to secure an organization, he must be aware of all potential threats, even if this threat is a ten-year-old protocol that represents less than 1 percent of the overall Internet traffic in 2008. Don’t be blinded by this 1 percent: This figure is doomed to increase in the coming years, and chances are good that your network is already exposed to some IPv6 threats. It’s better to be safe than sorry.

Just like the early deployment of many technologies, security is often left to the final stages of implementation. Our intent in writing this book is to improve the security of early IPv6 deployments from day one. Any organization considering or already in the midst of transitioning to IPv6 does not want to deploy a new technology that cannot be secured right from the outset. The transition to IPv6 is inevitable, and therefore this book can help you understand the threats that exist in IPv6 networks and give you ways to protect against them. Therefore, this book gives guidance on how to improve the security of IPv6 networks.

# Goals and Methods

Currently, many organizations have slowed their migration to IPv6 because they realize that the security products for IPv6 might be insufficient, despite the fact that the network infrastructure is ready to support IPv6 transport. They realize that they cannot deploy IPv6 without first considering the security of this new protocol. This book intends to survey the threats against IPv6 networks and provide solutions to mitigate those threats. It covers the issues and the best current practices.

This book is arranged so that it covers the threats first and then describes ways to combat these threats. By outlining all the risks and showing that a solution exists for each threat, you can feel more comfortable with continuing the transition to IPv6. You learn about techniques attackers might use to breach your networks and what Cisco products to use to protect the networks.

However, showing attacks without solutions is socially irresponsible, so the focus is on the current techniques that are available to make the IPv6 network more secure and on the best current practices.

By reading this book, you can gain an understanding of the full range of IPv6 security topics.

## Who Should Read This Book

This book is intended to be read by people in the IT industry who are responsible for securing computer networks. You should already know the basics of the IPv6 protocol and networking technology. This book is not an introduction to IPv6. There are many good books and online resources that can teach you about IPv6, and there are many great books on computer network security.

The intent of this book is to dive deeper into the protocol and discuss the protocol details from a security practitioner's perspective. It is a book for experts by experts. It covers the theory but at the same time gives practical examples that can be implemented.

## How This Book Is Organized

This book starts with a foundation of the security aspects of the IPv6 protocol. The early topics of this book are arranged from the outward perimeter of an organization's network inward to the LAN and server farms. The later chapters of the book cover advanced topics. This book can be read completely from start to finish; however, if you want to "skip around," that is fine. You should eventually read every chapter to gain a comprehensive knowledge of the subject matter.

Some of the information (such as tables and commands) in this book is for reference. You should refer back to this book when it comes time to implement. This gives you cookie-cutter examples to follow that should be in line with the best current practices for securing IPv6. However, do not just go through this book and implement every command listed. Perform some of your own basic research on these commands to make sure that they perform exactly what you intend your network to do.

IPv6 security is an incredibly active research area, and new protocols and new products will continually be developed after this book is written. It is our goal that the "shelf life" of this book is many years because the concepts will still be valid even as Cisco security products continue to evolve with the threat landscape. Every effort was made to make this book as current as possible at the time it was published, but you are advised to check whether new methods are available at the time of reading. The IPv6 security field is quickly evolving as IPv6 gets more widely deployed.

Chapters 1 through 12 cover the following topics:

- **Chapter 1, "Introduction to IPv6 Security":** This short chapter reintroduces IPv6, describes how widely it is deployed, discusses its vulnerabilities, and identifies what hackers already know about IPv6. Some initial mitigation techniques are presented.
- **Chapter 2, "IPv6 Protocol Security Vulnerabilities":** This chapter discusses the aspects of the IPv6 protocol itself that have security implications. Security issues related to ICMPv6 and the IPv6 header structure are covered. Demonstrations are conducted that show the protocol vulnerabilities, and solutions are given to mitigate those risks. This chapter also covers security issues of IPv6 network reconnaissance and address spoofing.
- **Chapter 3, "IPv6 Internet Security":** This chapter covers the large-scale threats against the IPv6 Internet and describes perimeter-filtering techniques that can help protect against those threats. Security for BGP peering is detailed in addition to other service provider–focused security practices. IPv6 MPLS security, security of customer equipment, IPv6 prefix delegation, and multihoming are reviewed.

- **Chapter 4, "IPv6 Perimeter Security":** This chapter covers the security threats that exist for perimeter networks that utilize IPv6. The chapter covers common filtering techniques that are deployed at the perimeter of the network. This chapter also covers IPv6 access lists, the IOS Firewall feature set, and the PIX/ASA/FWSM firewalls.
- **Chapter 5, "Local Network Security":** This chapter examines the threats against LANs. Many vulnerabilities exist on IPv6 access networks, and these vulnerabilities are covered along with many solutions for mitigating them. The chapter covers issues related to Neighbor Discovery Protocol, autoconfiguration addressing, and DHCPv6 communications on a LAN. This chapter also reviews SEND and describes how it can be implemented.
- **Chapter 6, "Hardening IPv6 Network Devices":** This chapter covers the security improvements that can be made to a network device running IPv6. Techniques for securing the management of network devices are reviewed. This chapter reviews ways to secure routing protocols and covers first-hop router redundancy protocols. Techniques for controlling the device's resources are detailed in addition to ways to control network traffic.
- **Chapter 7, "Server and Host Security":** This chapter covers the ways to secure a computer running IPv6. It is important to harden IPv6 nodes from the threats that exist. Microsoft, Linux, BSD, and Solaris operating system IPv6 security techniques are detailed. This chapter covers how host-based firewalls and Cisco Security Agent (CSA) can be used to protect IPv6 hosts.
- **Chapter 8, "IPsec and SSL Virtual Private Networks":** This chapter covers the basics of IPsec. The chapter reviews techniques for setting up site-to-site VPN links using IPv6, dynamic multipoint VPNs, as well as remote-access VPNs. The use of ISATAP over an IPsec client connection and the use of SSL VPNs with AnyConnect client are covered.
- **Chapter 9, "Security for IPv6 Mobility":** This chapter covers Mobile IPv6 and describes how securing this protocol can be challenging. Mobile IPv6 is reviewed, and the security implications are discussed. This chapter gives recommendations on how Mobile IPv6 can be used responsibly and safely. Additional IPv6-capable mobility solutions are covered along with their security implications.
- **Chapter 10, "Securing the Transition Mechanisms":** This chapter discusses the various techniques that are used to help organizations migrate from IPv4 to IPv6. Dual-stack, tunnel, and NAT migration techniques are covered along with their security issues. Each of these techniques has its own security implications and solutions for securing the traffic. This chapter covers the threats by showing examples of how an attacker might try to infiltrate a network. The security protections that can be used to keep the network safe during migration are also covered.
- **Chapter 11, "Security Monitoring":** This chapter covers the various systems that are currently available to monitor the security of IPv6 networks. Monitoring a network and the computers on the network is a critical aspect of any security practice. IPv6 networks are the same in this regard and must be managed appropriately. The topics of forensics, intrusion detection and prevention, security information management, and configuration management are covered.
- **Chapter 12, "IPv6 Security Conclusions":** This chapter summarizes the common themes discussed throughout the book. Commonalities between IPv4 security and IPv6 security are discussed. This chapter contains discussions about creating IPv6-specific security policies. This chapter also reviews what the future holds for IPv6 security. A consolidated list of IPv6 security recommendations is provided.

CHAPTER 3

# IPv6 Internet Security

Many people are surprised to learn that IPv6 is already running on the Internet. The Internet can run both IPv4 and IPv6 simultaneously because the protocols are independent of each other. Those who do not have IPv6 connectivity cannot access IPv6 services provided over the Internet.

There are many large-scale threats on the current IPv4 Internet, and IPv6 will be evaluated to improve this situation. These threats have the potential to deny service to critical services and spread malware. IPv6 can reduce many of the attacks that are so prevalent on the IPv4 Internet. Attackers can forge packets, so filtering based on IP address is a requirement. One of the key security measures when connecting to the Internet is to perform ingress and egress filtering of IPv6 packets. Because the IPv6 addresses are quite different than IPv4 addresses, filtering IPv6 addresses is also unique.

Security within a service provider's environment is also a focus area. How a service provider secures its network directly impacts the security of the Internet at large. Service providers use Border Gateway Protocol (BGP) extensively, so the secure use of this routing protocol is a fundamental practice. Service providers make use of Multiprotocol Label Switching (MPLS) in their core networks. This chapter covers the security of this protocol with respect to IPv6.

Service providers must connect millions of customers and their customer premises equipment (CPE) to the Internet. This must be done securely to provide worry-free Internet access to the general public. Because IPv6 addresses are assigned hierarchically, the assignment of addresses to customers must also be done safely.

Many enterprise customers want to be connected to multiple service providers for added assurance that their networks will remain operational if a single service provider's network has problems. However, this provides challenges for IPv6, so there are some emerging solutions to this conundrum.

This book starts out covering IPv6 security from the outside inward, so it is logical to start by looking at the Internet-facing network components. This chapter covers how to secure your network when it is connected to the IPv6 Internet.

# Large-Scale Internet Threats

The Internet is not a safe place anymore. Back in the late 1980s, the cooperative organizations that made up the Internet were primarily universities, research institutions, and military organizations. However, this changed on November 2, 1988, when the Morris Internet worm was unintentionally released. The Morris worm was the first large-scale Internet denial of service (DoS) attack. Until that time, the Internet was a communication tool for sharing information between collaborative and friendly organizations. After that event and as the Internet grew, the Internet started to have a sinister shadow that meant organizations connecting to the Internet needed to protect themselves.

Now that the Internet has evolved to use both IPv4 and IPv6, the threats have also evolved. Packet-flooding attacks are possible using either IP version. Internet worms operate differently in IPv6 networks because of the large address space. Distributed denial of service (DDoS) attacks are still possible on the IPv6 Internet, but there are some new ways to track them. This involves the use of tracing back an attack toward its source to stop the attack or find the identity of the attacker. The following sections cover each of these large-scale Internet threats and discuss prevention methods.

## Packet Flooding

IPv4 networks are susceptible to “Smurf” attacks, where a packet is forged from a victim’s address and then sent to the subnet broadcast of an IPv4 LAN segment (for example, 192.168.1.255/24). All hosts on that LAN segment receive that packet (icmp-echo with a large payload) and send back an echo reply to the spoofed victim address. This overloads the victim’s IP address with lots of traffic and causes a DoS. Many DoS attacks are easy to disable by simply entering **no ip directed broadcasts** to every Cisco Layer 3 interface within an organization. However, the default router behavior has been changed so now disabling directed broadcast forwarding is the default setting. This mitigation technique is documented in BCP 34/RFC 2504, “User’s Security Handbook.”

Because IPv6 does not use broadcasts as a form of communication, you might assume that these types of attacks are limited. However, IPv6 relies heavily on multicast, and these multicast addresses might be used for traffic amplification. An attacker on a subnet could try to send traffic to the link-local all nodes multicast address (FF02::1) and the link-local all routers multicast address (FF02::2).

One such example of using multicast to leverage an amplification attack is demonstrated with The Hacker’s Choice (THC) IPv6 Attack Toolkit. It contains two utilities named smurf6 and rsmurf6. They operate much the same as the original IPv4 Smurf attacks but instead use multicast to amplify the attack. The smurf6 tool sends locally initiated Internet Control Message Protocol version 6 (ICMPv6) echo request packets toward the multicast address FF02::1, and then the hosts on that LAN that are vulnerable to the attack generate ICMPv6 echo response packets back to the source, which is the unknowing victim. The smurf6 victim can be on the local subnet with the attacker or on a remote subnet.

Example 3-1 shows how smurf6 can be used to affect a computer on the same subnet as the attacker. If the victim is on a different segment, the systems on this segment send the echo replies to the remote victim's system. The first parameter is the local attacker's interface, and the second parameter is the victim's IPv6 address.

**Example 3-1** *Smurf6 Attack*

```
[root@fez thc-ipv6-0.7]# ./smurf6 eth0 2001:db8:11:0:b0f7:dd82:220:498b
Starting smurf6 attack against 2001:db8:11:0:b0f7:dd82:220:498b (Press Control-C to
  end) ...

[root@fez thc-ipv6-0.7]#
```

The rsmurf6 tool is coded a little differently. It sends ICMPv6 echo reply packets that are sourced from ff02::1 and destined for remote computers. If the destination computer (victim) is a Linux distribution that can respond to packets sourced from a multicast address, it responds to the source, which causes a traffic flood on the remote LAN. This form of amplification is particularly dangerous because each packet generated by rsmurf6 would translate into numerous packets on the remote LAN. Rsmurf6 is like a reverse smurf6 and only works on incorrectly coded implementations of the IPv6 stack. Therefore, it is not as effective as it once was when more vulnerable operating systems were in existence.

Example 3-2 shows how the rsmurf6 tool can be used. The first part of the example targets a victim's computer on a remote subnet. The second part of the example is destined for the link-local all nodes multicast address FF02::1 and essentially denies service to the entire local LAN that the attacker is connected to. Even the smallest systems can generate 25,000 pps, which is about 25 Mbps of traffic to all hosts.

**Example 3-2** *Rsmurf6 Attack*

```
[root@fez thc-ipv6-0.7]# ./rsmurf6 -r eth0 2001:db8:12:0:a00:46ff:fe51:9e46
Starting rsmurf6 against 2001:db8:12:0:a00:46ff:fe51:9e46 (Press Control-C to end)
  ...

[root@fez thc-ipv6-0.7]# ./rsmurf6 -r eth0 ff02::1
Starting rsmurf6 against ff02::1 (Press Control-C to end) ...

[root@fez thc-ipv6-0.7]#
```

It should be mentioned that these rsmurf6 attacks are only effective on computers that have IPv6 stacks that allow them to respond to an ICMPv6 packet that was sourced from a multicast address. Most modern IPv6 implementations are intelligent enough to recognize that this is not a valid condition, and they simply drop the packets. In other words, IPv6 hosts should not be responding to echo request packets destined to a multicast group address.

### More About ICMP and Amplification Attacks

RFC 2463, “Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification,” states that no ICMP messages can be generated in response to an IPv6 packet destined to a multicast group. The intent is to prevent all the amplification attacks if all IPv6 nodes correctly implement this RFC.

One issue with RFC 2463 is that there are two exceptions to the strict rule: “Packet too big” and “Parameter problem ICMP message” error messages can still be generated in response to a packet destined to a multicast group. This is required to allow path maximum transmission unit (MTU) discovery for a multicast video stream. This opens the door to an amplification attack in the same shot, even if all IPv6 nodes are RFC 2463 compliant.

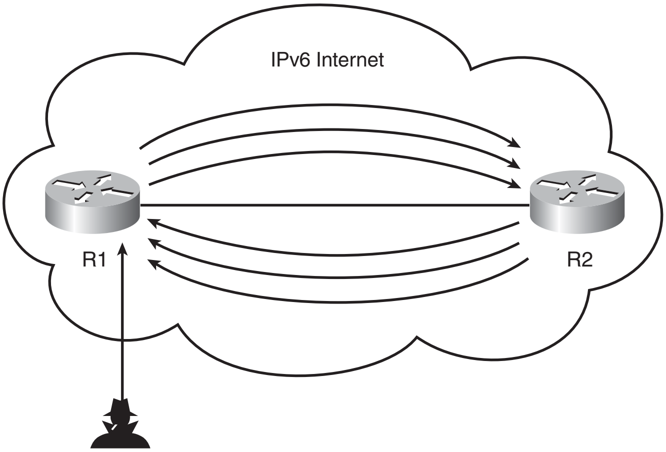
While the amplification attacks cannot be prevented at the node level, the effect can be thwarted by applying rate limiting to those ICMP messages: They should be rare in every network so that a rate limit (10 messages/sec) can permit the correct use of those messages (path MTU discovery) while blocking the amplification attack.

In Chapter 2, “IPv6 Protocol Security Vulnerabilities,” you learned that it is a good practice to limit who can send to multicast groups. Because IPv6 does not have broadcast as a form of communications, multicast is the method for one-to-many communications. For this reason, multicast can be leveraged by attackers for packet amplification attacks. Therefore, the solution is to tightly control who can send to multicast groups and when it is appropriate to respond to a multicast packet. Service providers can also consider rate-limiting user connections and particularly rate-limit IPv6 multicast traffic. Most multicasts should be confined to the LAN, so if an attacker is already on your LAN, you need to use other means to protect against that. Physical security, disabling unused switch ports, enabling Ethernet port security, and using an 802.1X or Network Admission Control (NAC) technology are options to prevent unauthorized access to the internal networks.

DoS attacks can be performed using a feedback loop to consume resources or amplify the packets sent to a victim. In Chapter 2, you saw how RH0 packets could be created with a list of embedded IPv6 addresses. The packet would be forwarded to every system in the list before finally being sent to the destination address. If the embedded IPv6 addresses in an RH0 packet were two systems on the Internet listed numerous times, it could cause a type of feedback loop.

Figure 3-1 shows how this type of ping-pong attack would work. The attacker would first send the crafted packet to a network device on the Internet that is susceptible. That system would forward it onto the next system in the list. The two systems could continue to do so until they ran out of bandwidth or resources. However, sometime soon, this type of attack will have limited success because RFC 5095 has deprecated the use of Type 0 routing headers in IPv6 implementations.

**Figure 3-1** *Internet Feedback Loop*



DoS attacks might not just be about flooding traffic. With IPv6, there are going to be a wider variety of nodes attached to the network. IPv6-enabled appliances, mobile devices, sensors, automobiles, and many others can all be networked and addressable. DoS attacks could simply target a specific model of device and render it inoperable. The results could be far more tragic if your IPv6-enabled automobile suddenly stops while on the autobahn. The benefits of using IPv6 are great but so are the consequences if the communication is not secured properly.

## Internet Worms

Worms are a type of attack that requires no human interaction. This is different than a virus, which usually requires some form of human interaction to activate. Worms spread by themselves, infect vulnerable computers, and then spread further. Worms perform the entire attack life cycle in one small amount of code. That small amount of code contains the instructions for reconnaissance of new systems, scanning for vulnerabilities, attacking a computer, securing its access, covering its tracks, and spreading further.

Worms can be affected by the introduction of IPv6. This new protocol can affect a worm's ability to spread. It can also affect the techniques that worm developers use to make their code propagate. There are already examples of worms that leverage IPv6. The following sections cover these topics and discuss ways to help prevent worms.

## Worm Propagation

Many of the widespread worms in the past eight years have leveraged some vulnerability in software running on a computer. Worms such as Code Red, NIMDA, MS/SQL Slammer, W32/Blaster, W32/Sobig, W32/MyDoom, W32/Bagel, Sasser, and Zotob all took advantage of some Microsoft service vulnerability. Some of them spread over the Internet, and some used email as the medium for reaching other systems. Many worms now spread through email (executable attachments, address books), peer-to-peer, instant message, or file sharing. These types of worm propagation techniques are unaffected by IPv6's introduction.

In the past, worms have used network scanning or random guessing to find other systems to spread to. Worms that spread to random IPv6 addresses cannot spread as fast as in IPv4 networks because IPv6 addresses are sparsely populated while IPv4 addresses are densely populated. Worms have been successful at scanning other IPv4 systems to infect because of the density of the current IPv4 space. Some worms have spread randomly (Code Red, Slammer), while others have spread sequentially (Blaster). It could be postulated that the Sapphire/SQL Slammer worm would not have been as successful on an IPv6 Internet because the size of the IPv6 address space is so large compared to IPv4. The Sapphire/SQL Slammer worm would take many thousands of years to reach its maximum potential on the IPv6 Internet. Given IPv6's immense address space, these types of worms will not be able to guess the addresses of other victims to spread to and infect. Random scanning will not be an option for worms on IPv6 networks. However, if IPv6 addresses are allocated sequentially or are otherwise densely packed, scanning can be just as fast as with IPv4.

## Speeding Worm Propagation in IPv6

As worms get smarter, they can overcome many of the issues related to scanning a large IPv6 address space. Worms can increase their scan rate to try to reach more hosts each second. IPv6 worms need to overcome the problems with performing reconnaissance on IPv6 networks. As discussed in Chapter 2, there are many places for a worm to look to help the worm find other hosts to spread to. Worms can also improve their knowledge of the population. This could be done by recognizing only the currently allocated IPv6 address blocks or by seeding their code with several vulnerable systems. Worms could also work to find new targets by looking at other sources of IPv6 addresses.

Worms could consult the infected computer's neighbor cache to find other local systems. The worms would also look anywhere IPv6 addresses are stored to help them identify new targets. Domain Name System (DNS) lookups, local DNS files, /etc/hosts, registries, SSH known_hosts, and other lists of hosts could be consulted. Worms might also listen to the LAN traffic to find other hosts. Sniffing neighbor solicitation packets, Duplicate Address Detection (DAD) packets, and routing updates would help them target specific populations of hosts rather than randomly scanning. Even information about IPv6 addresses stored in logs like syslog, /var/log/messages, and search engine logs would be valuable to a worm.

Worm developers will likely adjust their strategy for IPv6 networks. A worm could infect a single host, and then the worm could use that host's ability to send IPv6 multicast packets within the organization (for example, FF02::1, FF05::1, FF08::1). An example of this can be seen in "Windows Kernel TCP/IP/IGMPv3 and MLDv2 Vulnerability" (MS08-001, CVE-2007-0069), which was discovered early in 2008 (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-0069). This vulnerability leveraged a bug in the Windows multicast code using malformed Internet Group Management Protocol version 3 (IGMPv3) packets. A worm could leverage this vulnerability to attack nearby IPv6 hosts and spread to those infected computers. Therefore, a method for mitigating worm attacks could leverage the practice of constraining communication with IPv6 multicast addresses.

It is predicted that worms that check for routable address space can spread even faster. A worm could contain all the routable IP prefixes, and that list would help it eliminate "black" unallocated space. A worm could also look at a host's routing table or passively listen for routing updates (FF01::1 all routers multicast group) on a LAN to learn about other local networks to start scanning. For example, scanning could also be accelerated if the worm could perform a MAC address flood (CAM overflow attack) of the local LAN switch and then listen to all the packets.

Dual-stack worms could leverage either IPv4 or IPv6 protocols to spread in even faster ways than previously using only IPv4. However, with the density of the population using IPv4, worms could spread quickly over only IPv4. Some worms can use a dual-stack approach to infect systems rapidly over IPv4. The worms can check whether the system is dual-stacked and then perform a multicast probe. The systems that respond to the link-local multicast (FF02::1) are then attacked using IPv6. This technique could even accelerate worm propagation in the short term. However, eventually as more IPv6-only hosts exist, this technique will lose its effectiveness.

IPv6 worms must have more advanced techniques to overcome the problem of scanning IPv6 addresses to spread. As these worms are made more sophisticated, more code is required, and the size of the worm increases. This makes it more difficult for the worm to spread because the transmission of the worm requires multiple packets and slows the spread.

## Current IPv6 Worms

A few worms have already leveraged IPv6, and unfortunately there will be more in the future. The Slapper worm was released in 2002. It targeted Apache web servers on TCP port 80. After the worm attacked an Apache server, it would then create a copy and spread to other Apache web servers by randomly finding IPv4 servers. It had a sophisticated command and control channel that would allow a hacker to create send commands to the infected servers. One command would send a flood of IPv6 packets toward a victim. Slapper was the first worm that had any type of IPv6 component to it.

W32/Sdbot-VJ is a spyware worm that tries to use the popularity of IPv6 to disguise itself. It does not use IPv6 to spread to other machines; however, it installs the program wipv6.exe and installs several registry entries. The user might be hesitant to delete the file because it might have something to do with the Windows IPv6 drivers. Therefore, it was less likely to be deleted from a computer.

### Preventing IPv6 Worms

A few techniques can help contain IPv6 worms. You must keep your antivirus and intrusion prevention system (IPS) signatures up to date so that they can identify new threats. Many worms leverage recent vulnerabilities that have been patched by the manufacturer, but not all customers have implemented the patch. Therefore, keeping software patched on computers and servers is a must. You can also use anomaly detection systems to identify an abnormal spike in traffic of any single protocol type. This would be one way to detect a problem, but the quicker you can detect a rapidly spreading worm and respond to block the propagation, the easier your remediation.

## Distributed Denial of Service and Botnets

Sophisticated hackers try to strive for elegant attacks that satisfy their need to prove their superiority. However, many times an advanced attack is not possible and an attacker might still want to perform some type of disruption. Oftentimes it is the less-experienced attackers that simply try to negatively impact a site after they fail at a more sophisticated attack. When their attempts are thwarted, they fall back to trying to cause damage by simply breaking the system and taking it offline. This attack performs a DoS and makes the system unable to provide service to the legitimate users. Attacks of this style that involve a large number of geographically disperse computers are called distributed denial of service (DDoS) attacks.

DDoS attacks are performed by a large set of many Internet-connected computers that have been compromised. These large numbers of computers are controlled by other compromised systems called handlers. The hacker that controls all these computers can send commands to their vast army of “zombies” to send traffic to a victim. These zombie computers are typically Internet-user PCs that have been turned into robots (bots for short) through malicious software. When the “bot herder” directs the botnet to send the large volume of traffic toward the victim, it prevents the victim from being able to communicate. Thus the attack denies the victim Internet access or denies the user’s access to the victim’s website.

## DDoS on IPv6 Networks

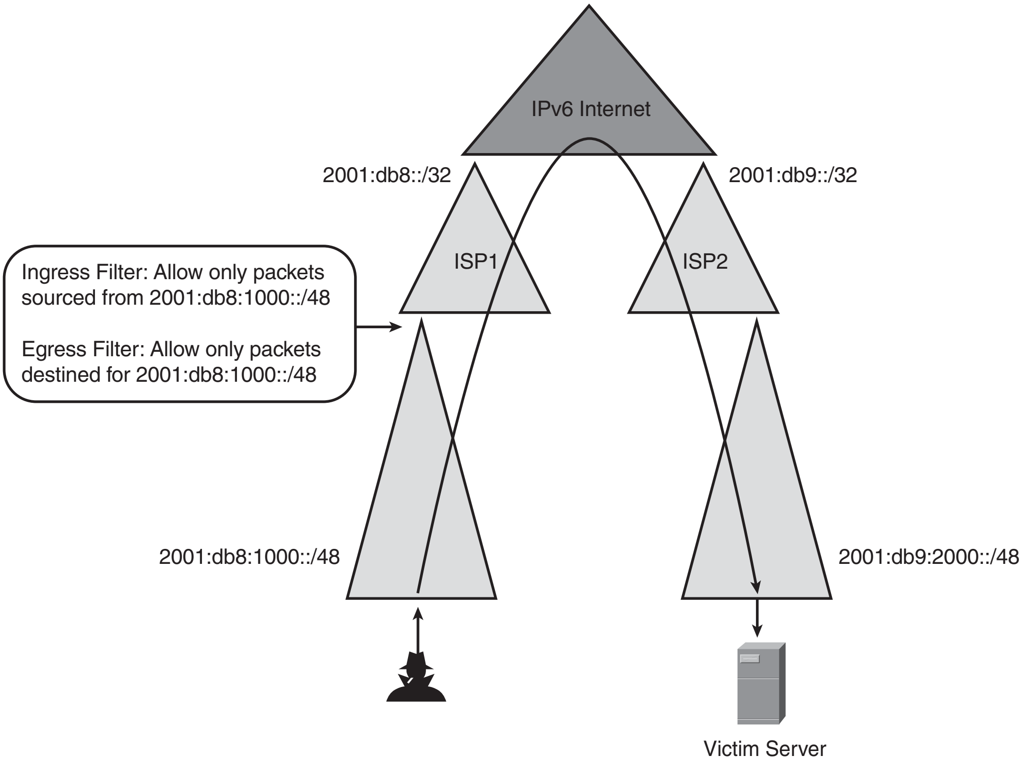
DDoS attacks can exist on an IPv6 Internet just like they exist on the current IPv4 Internet. Botnets, which are large networks of zombie infected computers, can be created, and their attacks can be focused on a victim. The use of IPv6 will not change the way that botnets are created and operated. DDoS botnets will unfortunately still exist on IPv6 networks. Botnets can also be used to send email spam and conduct other types of mischief. IPv6 will allow the Internet to contain many more devices than the IPv4 Internet. Imagine if many of these devices were to launch a DDoS attack. The results could be more devastating than today's attacks on the IPv4 Internet.

## Attack Filtering

Because an IPv6 address is allocated in a fully hierarchical manner, it would be easier to track down where the traffic is coming from and going to than on the IPv4 Internet, where addresses are not hierarchical. Because of fully hierarchical addressing, inbound/outbound source IP address filtering and unicast Reverse Path Forwarding (RFP) checks will be possible. Viruses and worms that spread using spoofed source addresses will be limited in an IPv6 network if Unicast RPF checks are deployed. Ingress and egress filtering will also limit these types of attacks.

Figure 3-2 shows how two Internet service providers (ISP) have assigned address space to two organizations. If one organization connected to ISP1 sends a large volume of traffic to the victim's host, it could be filtered by ISP1. The traffic could be validated to have legitimate source addresses coming from its assigned address space. Packets with spoofed source addresses would not be allowed to leave the organization. Therefore, if the victim saw attack traffic coming from the 2001:db8:1000::/48 address space, it could be traced back to its source. If an attacking host was using privacy addressing for the network ID portion of the address, the attack could only be traced back as far as the organization.

**Figure 3-2** *Internet Ingress/Egress Filtering*


IPv6 Internet
2001:db8::/32
2001:db9::/32
ISP1
ISP2
Ingress Filter: Allow only packets sourced from 2001:db8:1000::/48
Egress Filter: Allow only packets destined for 2001:db8:1000::/48
2001:db8:1000::/48
2001:db9:2000::/48
Victim Server

The hope is that if all ISPs and end-user organizations were to implement full ingress and egress address-spoofing filtering, this would help with tracking down the DDoS attacks. The infected computers could then be quickly determined, and the malicious software could be remediated more quickly.

## Attacker Traceback

In the unfortunate circumstance where you have fallen victim to a DoS attack, your first instinct is to look upstream for assistance. The goal is to try to identify the source of the traffic that is coming your way and stop it as close to the source as possible. You must coordinate with your ISP to help contain a DoS attack. Your organization should not wait until this happens to work out procedures with your ISP to help you handle this. An organization should know ahead of time the contact information and procedures to follow to perform last-hop traceback.

Traceback in IPv6 networks involves finding the source address of the offending packets and then tracking down the offending host to a subnet. Then, tracking down the IPv6 address and the binding to the Layer 2 address (or asymmetric digital subscriber line [ADSL] port) of the host can be done at that site. Then, one could find out what Ethernet port the user is connected to and then investigate further. This procedure should be documented ahead of time so that it can be used quickly during an attack.

This process is time consuming and takes coordination between your own ISP and many others, and it is not applicable if the attack is a DDoS because there are literally thousands of attack sources. If you are trying to stop an attack by a botnet that could potentially contain thousands of bots, the task is overwhelming. Each of these bots is not sending traffic sourced from its own IP address, so tracing back to this many systems seems futile. The zombie hosts create traffic that looks like normal web traffic, so finding out which connections are legitimate is nearly impossible. The traffic patterns that these botnets create can be observed by using NetFlow to track statistics about each protocol flow. The flow records can be checked for traffic coming to or from an organization or service provider network. The collected NetFlow data can help trace the source of the traffic back to the source organization's network. However, the act of reaching out to that many users to have them remediate their systems is not feasible in most situations.

Your organization probably has a firewall, and you might have an IPS. Those two systems can try to stop the attack by filtering out traffic. However, the web requests will not match any known "signature," but either system can easily be configured to simply drop all traffic. That can stop the attack, but it would also stop all other valid users from reaching your servers. Furthermore, your Internet connection can be so saturated with traffic that blocking at your site has limited value.

If the attack that is hitting your network is a SYN-Flood attack, a solution is available. A SYN-Flood attack is where the packets with spoofed source addresses are sent to the web servers and they have the SYN TCP flag set. The server tries the second part of the three-way TCP handshake by sending back a SYN-ACK TCP flag packet to the spoofed source address. Because that packet never reaches the spoofed source, the three-way handshake never takes place and the web server retains the state of the connection for some time. Meanwhile the web server is hit with many of these false connections, and they drive up its CPU and memory utilization.

A technique that would help in this instance is to leverage an application front-end system or server load-balancing system that can terminate those SYN packets and send back the SYN-ACK on behalf of the server. The SYN cookie technique can also be used to verify the initial sequence number (ISN) of the client connection. If the client sends back the legitimate final ACK to complete the three-way handshake, the connection is legitimate. The server load balancer can then make the connection to the web server on behalf of the client, and the HTTP request can take place normally. False SYN-Flood traffic does not reach the server, but legitimate connections are served.

## Black Holes and Dark Nets

During any type of attack or for other reasons an ISP can create a situation where traffic destined for a site can be dropped. The traffic is routed into a black hole, where it is simply discarded. To do this, the service provider creates a route to Null 0 on its routers and redistributes that route to the other peering routers in its infrastructure. The route can be for an entire prefix or for a specific IP address. All the routers with this null route simply drop the packets destined for that prefix. This technique was defined in RFC 3882, "Configuring BGP to Block Denial-of-Service Attacks," and is also known as a *Remotely Triggered Black Hole (RTBH)*. The problem with this technique is that it is crude and can block legitimate traffic as well as the malicious traffic from the attack. However, this same technique can be applied to the IPv6 Internet.

ISPs also can use the RTBH technique to trace the source of the malicious traffic. When the traffic is routed to the black hole, ICMP error messages are created. Monitoring the ICMP error messages gives an indication of where the traffic entered the service provider's network. There are many different versions of this same technique. Different ISPs use different solutions to help them track down where the malicious traffic is entering their network. The goal is to identify where the traffic is coming from and then work back toward the source. This usually involves cooperation with other ISPs.

Another technique for learning about Internet threats involves the creation of a darknet for some portion of public address space. A public prefix is advertised by a service provider to the Internet, but that prefix has no services within it. Instead that network contains a computer that is monitoring all traffic coming into that network. Any packets that are on the service provider's network destined for that address space end up being monitored. Because that prefix has never been used, there is no legitimate reason for any packets to be going to it. Therefore, the only things going to the darknet network are transient packets that can be the results of scanning attacks.

Darknets, or network telescopes as they are also known, help researchers understand hacker behavior. They are similar to a honeynet, but there is no interaction with the hacker. No packets leave the darknet, but anything that enters the darknet is seen by a protocol sniffer. The sniffer can archive the data for future analysis and it can also pick up trends. However, few packets enter an IPv6 darknet, so it can be difficult to interpret results. However, there is a lot of public IPv6 address space available to perform these types of experiments.

**NOTE** The book *Router Security Strategies: Securing IP Network Traffic Planes*, by Gregg Schudel and David J. Smith (Cisco Press, 2008), describes the preceding techniques in more detail.

# Ingress/Egress Filtering

One of the important aspects of perimeter security is filtering at an organization's borders. If you are a service provider, your network borders are customers and other service providers. If you are an enterprise, your network borders are ISPs and other business partner organizations. There are commonalities in the filtering of route advertisements done by service providers and the route filtering done by their customers. One key difference involves the way IPv6 routes are filtered at the Internet's edge. One commonality is the filtering of bogus addresses that should not be used in either the source address or the destination address header field. The following sections describe the different methods of filtering routes and give example of how to filter allocated and bogus IPv6 address prefixes.

## Filtering IPv6 Traffic

Service providers typically do not filter individual customer packets traversing their networks based on the packet's contents. However, they should help protect the Internet and their own infrastructure by performing filtering at their perimeters. BCP 84/RFC 3704, "Ingress Filtering for Multihomed Networks," (Best Current Practice [BCP] 84) covers the practice for IPv4 networks. Now these same principles can be extended to IPv6 networks.

Performing IPv6 traffic filtering for high-speed links would require systems that can perform filtering in hardware. Service providers could also filter packets that do not conform to the IPv6 specifications. The points where a service provider network touches customers and other providers are locations where the filtering should occur. This type of filtering is not done by firewalls on the traffic itself but rather on the routing update exchanges.

## Filtering on Allocated Addresses

With IPv4, customers can get address allocations from their provider and also obtain their own address space. In IPv6, the intent is to require all customers to get their allocations from their service provider. The service providers receive their addresses from the Regional Internet Registries (RIR), who in turn receive their allocations from the Internet Assigned Numbers Authority (IANA). This creates a fully hierarchical addressing structure that maximizes the use of aggregation and is sure to reduce the size of the Internet routing table. RIRs can also assign provider independent (PI) address blocks to customers. However, these blocks might not be allowed to be routed on the Internet, even if it can be expected that more and more ISPs will have to allow the transit traffic destined to PI addresses.

ISPs need to be careful about the address space that they are using and assigning to customers. Filtering what you are advertising and what you are receiving over peers also helps prevent many types of BGP threats. Receiving more-specific routes, less specific routes, routes for unallocated space, and malicious routes are threats that can all be prevented through careful filtering of routes. Receiving many of these different types of routes can either be accidental or malicious on the customer's part, and you might not know which. Being overly permissive on the types of routes allowed to be advertised to the ISP from customers is not wise. Distribute lists, prefix lists, and route maps can all be used to control what routes are being sent and received.

You might not want to accept more specific routes from customers or peers because that could be one way that an attack takes place. Because the minimum allocation size is a /48, service providers might also want to simply reject any /49 or longer prefix. Therefore, you might not want to accept a BGP advertisement with anything smaller than a /48, regardless of the prefix. BGP also makes the assumption that a peer has the authority to advertise the prefix and autonomous system (AS) paths. If these are falsified, all types of routing instability can occur.

ISPs have the responsibility to perform careful filtering of customer routes. There are many address blocks that a service provider should not receive from a customer or a peer. The ISP must also allow the customer to be able to route its traffic to and from the Internet. These customer routes must be filtered at the point where the two networks meet. It is also a good practice for the service provider to check the regional registry to make sure that the customer is the rightful owner of the prefix. This can be done with whois information from the Shared WHOIS Project (SWIP). For example, if a customer is assigned the address block 2001:db8:100::/48, the inbound prefix list permitting this advertised route would look like the configuration shown in Example 3-3. This example shows a prefix list that would allow only the customer's block and nothing else.

**Example 3-3** *Filtering Customer Address Assignment*

```
ipv6 prefix-list v6-cust-routes permit 2001:db8:100:100::/48
ipv6 prefix-list v6-cust-routes deny ::/0 le 128
!
route-map CUSTROUTES permit 10
 match ipv6 address prefix-list v6-cust-routes
!
router bgp 100
 neighbor 2001:db8:100:100::1 remote-as 200
 neighbor 2001:db8:100:100::1 route-map CUSTROUTES in
```

You should disallow overly specific prefixes and disallow any prefix greater than /48. The more-specific /64 route for the customer network is quelled while the aggregate /48 is advertised. Some ISPs can elect to allow more-specific routes from customers, but they should not be smaller than a /48.

## Bogon Filtering

Bogons are the IP address ranges that either have not been allocated or are reserved. The word *bogon* is a derivative of the word bogus, which means illegitimate or fake and is similar to terms for subatomic particles used in quantum mechanics. The bogons list originated from RFC 3330's list of "Special-Use IPv4 Addresses," and now a similar list of "Special-Use IPv6 Addresses" is documented in RFC 5156. Packets with these addresses, either used as source addresses or destination addresses, should not be routed on the Internet. These are often blocked at IPv4 routers explicitly because there are a finite number of these. Lists are maintained that contain the IPv4 address space, and service providers and other organizations use these bogon lists. The bogon lists help to craft filters to prevent these packets from traversing network perimeters.

The list of valid IPv6 address blocks is maintained by the IANA. This list shows the address space allocations and the organizations responsible for maintaining that address space. At the time that this book is written, the current allocations are listed at the following URL:

http://www.iana.org/assignments/ipv6-unicast-address-assignments

The IANA has also made special registrations of address spaces for specific purposes. This is done because there are times when addresses are required for a specific purpose, but these addresses will not be allocated to an organization. The IANA Special Purpose Address Registry is defined by RFC 4773, "Administration of the IANA Special Purpose IPv6 Address Block," and is available at http://www.iana.org/assignments/iana-ipv6-special-registry.

In general, you should always filter packets coming to you that are sourced from bogon addresses. This is a good goal, but it also means that you need to stay on top of the allocations as they are made and adjust the filter lists accordingly. These bogon lists can change several times each year.

You should also take into consideration the address space that you have been allocated as a service provider. Service providers have out-of-band management networks. Filtering these internal addresses at the borders of the service provider can help prevent attacks against the back-office/internal systems (that is, billing, management, and so on). You should filter the infrastructure addresses that are used by your network equipment and router interfaces. Therefore, you must filter packets coming to you from your own allocated address space. This can be done at your network perimeters with the use of Unicast Reverse Path Forwarding (Unicast RPF) checks. You should also deny your own allocated address space from being advertised to you from a customer or any peer. You know about your addresses, and you should not let anyone tell you any differently. That should protect anyone from trying to destabilize your routing.

Many other prefixes should be denied inbound and outbound at your network perimeters. Table 3-1 gives the list of the routes that should be filtered from entering or leaving your network. These should not be advertised to you from any customer or peer, and you should also prevent yourself from advertising these.

**Table 3-1** *Prefixes That Should Be Blocked*

| Routes to Block | Prefixes |
|---|---|
| Default route | ::/0 |
| Unspecified address | ::/128 |
| Loopback address | ::1/128 |
| IPv4-compatible addresses | ::/96 |
| IPv4-mapped addresses | ::ffff:0.0.0.0/96 |
| Link-local addresses | fe80::/10 or longer |
| Site-local addresses (deprecated) | fec0::/10 or longer |
| Unique-local addresses | fc00::/7 or longer |
| Multicast addresses | ff00::/8 or longer |
| Documentation addresses | 2001:db8::/32 or longer |
| 6Bone addresses (deprecated) | 3ffe::/16 |

Some of the entries in Table 3-1 can be covered with a single prefix. For example, unspecified routes, loopbacks, and IPv4-mapped addresses can all be matched with 0000::/8 or longer.

Because so little of the IPv6 address space has been allocated, it is easier to permit the legitimate route addresses than to try to deny all the routes that should be blocked. Therefore, route filters have permit statements for the legitimate prefixes, and all other routes are blocked by the implicit deny-all at the end of the list. Therefore, the list of allocated IPv6 addresses can be specified within an IOS prefix list and applied to the external interface of an Internet router. Example 3-4 shows an example of this prefix list and indicates how it can be applied to a BGP peer. This filter list comes from the Team Cymru IPv6 bogon filter list for Cisco IOS routers: http://www.cymru.com/Bogons/v6ios.html.

**Example 3-4** *Bogon Prefix Filter List*

```
ipv6 prefix-list ipv6-global-route deny   2001:0DB8::/32 le 128
IPv6 prefix-list IPv6-global-route deny   <your own allocated addresses>/32
ipv6 prefix-list ipv6-global-route permit 2001:0000::/32
ipv6 prefix-list ipv6-global-route permit 2001:0200::/23 ge 23 le 64
ipv6 prefix-list ipv6-global-route permit 2001:0400::/23 ge 23 le 64
ipv6 prefix-list ipv6-global-route permit 2001:0600::/23 ge 23 le 64
ipv6 prefix-list ipv6-global-route permit 2001:0800::/23 ge 23 le 64
```

**Example 3-4** *Bogon Prefix Filter List (Continued)*

```
ipv6 prefix-list ipv6-global-route permit 2001:0A00::/23 ge 23 le 64
ipv6 prefix-list ipv6-global-route permit 2001:0C00::/23 ge 23 le 64
ipv6 prefix-list ipv6-global-route permit 2001:0E00::/23 ge 23 le 64
ipv6 prefix-list ipv6-global-route permit 2001:1200::/23 ge 23 le 64
ipv6 prefix-list ipv6-global-route permit 2001:1400::/23 ge 23 le 64
ipv6 prefix-list ipv6-global-route permit 2001:1600::/23 ge 23 le 64
ipv6 prefix-list ipv6-global-route permit 2001:1800::/23 ge 23 le 64
ipv6 prefix-list ipv6-global-route permit 2001:1A00::/23 ge 23 le 64
ipv6 prefix-list ipv6-global-route permit 2001:1C00::/22 ge 22 le 64
ipv6 prefix-list ipv6-global-route permit 2001:2000::/20 ge 20 le 64
ipv6 prefix-list ipv6-global-route permit 2001:3000::/21 ge 21 le 64
ipv6 prefix-list ipv6-global-route permit 2001:3800::/22 ge 22 le 64
ipv6 prefix-list ipv6-global-route permit 2001:4000::/23 ge 23 le 64
ipv6 prefix-list ipv6-global-route permit 2001:4200::/23 ge 23 le 64
ipv6 prefix-list ipv6-global-route permit 2001:4400::/23 ge 23 le 64
ipv6 prefix-list ipv6-global-route permit 2001:4600::/23 ge 23 le 64
ipv6 prefix-list ipv6-global-route permit 2001:4800::/23 ge 23 le 64
ipv6 prefix-list ipv6-global-route permit 2001:4A00::/23 ge 23 le 64
ipv6 prefix-list ipv6-global-route permit 2001:4C00::/23 ge 23 le 64
ipv6 prefix-list ipv6-global-route permit 2001:5000::/20 ge 20 le 64
ipv6 prefix-list ipv6-global-route permit 2001:8000::/19 ge 19 le 64
ipv6 prefix-list ipv6-global-route permit 2001:A000::/20 ge 20 le 64
ipv6 prefix-list ipv6-global-route permit 2001:B000::/20 ge 20 le 64
ipv6 prefix-list ipv6-global-route permit 2002:0000::/16 ge 16 le 64
ipv6 prefix-list ipv6-global-route permit 2003:0000::/18 ge 18 le 64
ipv6 prefix-list ipv6-global-route permit 2400:0000::/12 ge 12 le 64
ipv6 prefix-list ipv6-global-route permit 2600:0000::/12 ge 12 le 64
ipv6 prefix-list ipv6-global-route permit 2610:0000::/23 ge 23 le 64
ipv6 prefix-list ipv6-global-route permit 2620:0000::/23 ge 23 le 64
ipv6 prefix-list ipv6-global-route permit 2800:0000::/12 ge 12 le 64
ipv6 prefix-list ipv6-global-route permit 2A00:0000::/12 ge 12 le 64
ipv6 prefix-list ipv6-global-route permit 2C00:0000::/12 ge 12 le 64
!
router bgp 64500
 neighbor 2001:db8:1::2 route-map ACCEPT-ROUTES in
!
route-map ACCEPT-ROUTES permit 10
 match ip address prefix-list ipv6-global-route
```

**NOTE** These lists must be updated as soon as new allocations are made. This means following the IANA and the regional registry websites, mailing lists, and changes to these filters. There are groups such as Team Cymru that also maintain up-to-date lists and examples of filters. Otherwise new customers who might have received an allocation from one of these new blocks must troubleshoot why their packets are being blocked to and from various places on the Internet. The Team Cymru IPv6 bogons list can be found at http://www.cymru.com/Bogons/v6bogon.html.

## Bogon Filtering Challenges and Automation

Filtering what prefixes are advertised by an end-user organization is a best practice. It is also a best practice to filter prefixes from a service provider's other service provider peers. Most peers just permit the /32s that other peers have been allocated. Many service providers trust the peers they connect to and do not perform the necessary filtering to protect the Internet from dramatic problems. These service providers know that filtering bogons from being advertised to them is the right thing to do. However, many service providers cite the fact that bogon filtering can be hard to maintain because it is likely to change. Some service providers manually configure bogon filters, but the updating of the configurations can be automated with some form of script. In fact, when new address space is allocated by the IANA or the registries, the address space is usually given to Tier 1 ISPs because they will start to route the traffic appropriately for their customers.

There are techniques that service providers can use to help alleviate the burden of maintaining peer filters. It is easy to set up an automated method of updating the bogon list on all peering routers. After the filter is updated, you do not need to reset the peer to have the filter activate. When the peers are reset softly or the route flaps, the updates show up in the routing table.

Another technique for filtering routes to a peer is to leverage an Internet Routing Registry (IRR). These databases contain the registered address allocations for other ISPs, and they can help you create the prefix list applied to that peer. Routing Policy Specification Language (RPSL) is defined in RFC 2622 as a language to send and receive information from a registry. Recently, RPSLng (RFC 4012) added IPv6 and multicast support to its set of classes of objects. For example, one of the RPSL classes is called the ROUTE6 object, which contains the identification of the /32 addresses that service providers have been allocated. With objects like this, an IRR can be used to create a specific import or export route filter for the prefixes that should be sent or received from a peer. This would add to the security of IPv6 because filters could be automated and based on accurate sources of allocated and assigned prefixes. For these reasons, the IRRs must be secured, and the validity of the data must be regularly checked.

The historical challenges with IRRs were that the information was not accurate. Because the IPv6 Internet is in its early stages and the current Internet IPv6 routing table has few entries, the data will be easy to validate. Currently the set of IPv6 information in the IRRs would be small and easy to start a clean slate and maintain it. IRRs can help avoid mistakes made by humans and speed deployment through automation. Automation tools exist for IRRs (IRRToolSet, IRR Power Tools) to help create filters for peers and customer connections.

# Securing BGP Sessions

The Border Gateway Protocol version 4 (BGP4) protocol has been in existence since 1994 and has been updated several times over the past 15 years. BGP4, defined in RFC 4271, is the routing protocol used between autonomous systems that make up the Internet. External BGP (EBGP) is used between autonomous systems, and Internal BGP (IBGP) is used within an autonomous system. BGP is a path-vector routing protocol, where the paths are the list of autonomous systems that must be traversed to reach the destination prefix. Through the years, BGP has been extended to carry different types of routing information. RFC 4760, "Multiprotocol Extensions for BGP-4," allows BGP to operate over IPv4 or IPv6 and carry either type of routing information.

BGP is the central nervous system with which virtually all service providers are wired. Because BGP is the critical routing protocol of the Internet, it is a target of attacks. Attackers know that if they can find a weakness in BGP and exploit it, they could potentially destabilize the entire Internet. RFC 4272, "BGP Security Vulnerabilities Analysis," showed the weaknesses in BGP that service providers should try to prevent. Therefore, it is important that you work to secure BGP by focusing on the following areas:

- **Authentication:** Who are you talking to?
- **Confidentiality:** How do we communicate?
- **Integrity:** What is being said?
- **Availability:** Are you there?

Conventionally there are several approaches to securing BGP sessions, including the following:

- Explicitly configured BGP peers
- Using BGP session shared secrets
- Leveraging an IPsec tunnel
- Using loopback addresses on BGP peers
- Controlling the Time-to-Live (TTL) on BGP packets
- Filtering on the peering interface
- Using link-local peering
- Preventing long AS paths
- Limiting the number of prefixes received
- Preventing BGP updates that contain private AS numbers
- Maximizing BGP peer availability
- Logging BGP neighbor activity
- Securing IGP
- Extreme measures for securing communications between BGP peers

The following sections briefly describe each of these methods. More extreme measures that are not frequently used are also briefly mentioned later in this chapter.

## Explicitly Configured BGP Peers

One technique for securing BGP sessions is the concept that BGP sessions must be configured on each peering router. Peering is done explicitly by both BGP speakers. Therefore, a router will not form a peering session with another router that it has not been configured to peer with, and both peers mutually agree upon the BGP settings. A BGP peering session is not established if only one router is configured. There must be complementary configurations on each side for communications to take place. BGP communications take place over TCP, so the protocol must rely on a properly configured IP-layer foundation. BGP uses TCP port 179, so it has some inherent security in the fact that it is a connection-oriented protocol. TCP session state is maintained between the two peers.

The fact that BGP is a stateful transport layer routing protocol would normally provide some level of security, but it is also one of BGP's weaknesses. Attackers can spoof BGP packets and send them toward one of the BGP routers, or they could attack the TCP peering session between two BGP routers. Threats against long-lived TCP sessions involve TCP session hijacking using sequence number predication to reset one of the peers. One solution to this problem is to have BGP implementations use strong sequence number randomization. Therefore guessing the next sequence number or acknowledgment (ACK) number would be difficult and improbable.

## Using BGP Session Shared Secrets

One of the most widely used methods of securing BGP communications is to use a shared secret (password). RFC 2385, "Protection of BGP Sessions via the TCP MD5 Signature Option," defines how a simple password can be used with a message digest algorithm 5 (MD5) digest inserted into the BGP packets. This digest adds authentication to BGP and helps prevent an attacker from spoofing a BGP peer.

Even though it is a best practice to use a different password for every peering session, this can be difficult to maintain. Regardless, it is unwise to use the same secret password for all peering sessions. As they say, it is not a secret if you tell a bunch of people. RFC 3562, "Key Management Considerations for the TCP MD5 Signature Option," defines how a centralized system can maintain the security of the keys for all organizations. On a Cisco router, the password is assigned at the time that the neighbor is configured. Following is the router configuration command to enable MD5 authentication for a BGP peer:

```
neighbor neighbor-ipv6-address password P@ssw0rd
```

## Leveraging an IPsec Tunnel

Another technique for securing BGP communications is to leverage the security of an IPsec tunnel. IPsec is a strong way to secure BGP peers, protect the integrity of updates, and assist in preventing DoS attacks that target BGP peers. Using IPsec is better than MD5 because it keeps the keys refreshed over time. Because BGP is a TCP protocol, it can use IPsec with no modification. However, an IPsec connection must be created for the peering to form. This can add significant overhead to the routers, so it might be prohibitive in terms of CPU resources. Configuring and troubleshooting the IPsec tunnel can add significant burden to maintaining a service provider network. Furthermore, the IPsec tunnel that is used for sending routing information is thus used to forward traffic. The added packet-size overhead that IPsec adds would negatively impact throughput performance. Even though using IPsec is a secure method, it is not widely used.

Even still, an attacker who knows that a router is using authentication can simply create a large number of spoofed packets with fake authentication parameters and send them toward that router. This would cause the router to process these fake packets (even if they are quickly rejected) and artificially consume router resources. The CPU spike on the target router could delay legitimate routing traffic, thus accomplishing the attacker's goal of disrupting a network. Attackers could launch many authentication failures at the BGP router to potentially crash it. Therefore, authentication cannot be the only method of securing BGP communications.

Other methods of preventing unwanted traffic coming toward a router from causing problems involves filtering with access control lists (ACL). Control Plane Policing (CoPP) or Control Plane Protection (CPPr) can filter packets on the control plane of the router. Infrastructure ACLs (iACL) and receive ACLs can prevent the undesirable packets from reaching the router in the first place. Both of these concepts are covered fully in Chapter 6, "Hardening IPv6 Network Devices. "

## Using Loopback Addresses on BGP Peers

By using loopback addresses to peer BGP routers, it is more difficult for an attacker to know the source address of the TCP 179 peering session if the IP address could not be determined through the use of traceroute. Because loopbacks are logical interfaces, peering with loopbacks makes the BGP peers less physically connected and requires an Interior Gateway Protocol (IGP). Loopback interfaces are always up and operational, so they are very stable interfaces for the router to source many types of communications such as authentication, authorization, and accounting (AAA) or management traffic. Peering between loopback addresses is more popular on IBGP peers than EBGP peers because IBGP connections rely on an IGP. EBGP peers typically use the directly connected IP addresses on each end of the physical link, but these addresses can be easily discovered by attackers. Regardless, having a loopback IPv4 address as the router ID (RID) for the BGP process is a best practice.

## Controlling the Time-to-Live (TTL) on BGP Packets

Another technique involves controlling the TTL value that is set in the IP header on the TCP port 179 packets. EBGP routers send updates with a TTL typically set to 255, and EBGP routers typically accept packets that have a TTL set to 0 or greater. The problem is that an EBGP router can accept BGP packets that could have surreptitiously come from a network many hops away. If the TTL is constrained so that the TCP packets cannot travel beyond the direct physical connection between two peers, some security is gained. IBGP routers typically peer over many physical hops, so this technique is not necessarily applicable in all situations.

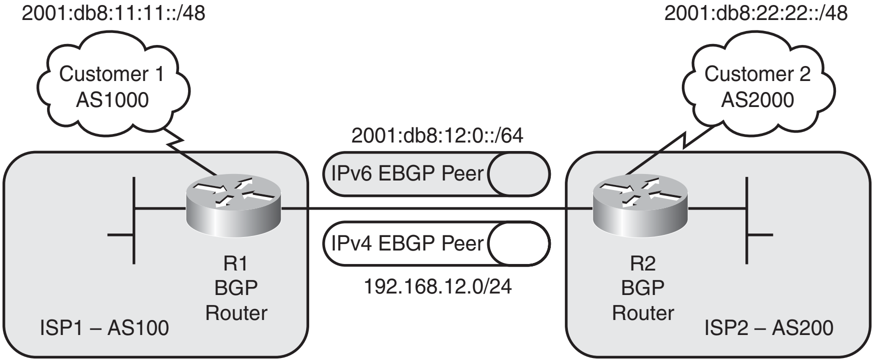
To secure EBGP peers and create a better TTL algorithm, the Internet Engineering Task Force (IETF) devised BGP TTL Security Hack (BTSH), which is also known as the Generalized TTL-based Security Mechanism (GTSM) (RFC 3682). This technique makes the EBGP router send TCP 179 packets to its peer with the TTL set to 255. The remote peer receives the BGP packet, and the router decrements the TTL to 254. That remote EBGP peer can then only accept BGP packets that have a TTL set to 254 or higher. This enforces the rule that EBGP peers only accept BGP packets from the directly connected peers that are only one hop away. If a spoofed BGP peer sending BGP packets comes from two hops away, the targeted router receives a TTL of 253. Because this TTL value of the forged packet is not greater than 254, that packet fails the test and is silently discarded. Therefore, packets with TTL values lower than 254 have originated more than one hop away. The TTL settings need to be configured on both peers to be effective. BTSH was first available in Cisco IOS Releases 12.0(27)S, 12.3(7)T, and 12.2(25)S. This technique is also affectionately referred to as the TTL-Hack. Following is the command that is used on each neighbor:

```
neighbor neighbor-ipv6-address ttl-security hops 1
```

BTSH helps with attacks against BGP, but it is not a complete solution within itself. For example, BTSH is not available to use on IBGP sessions. In addition to several other combinations, the TTL-Hack is a stronger strategy. It should also be mentioned that MD5 passwords and the TTL checking are both handled by the router CPU. These might be stronger techniques if routers start to support these security measures in hardware.

You can configure a reasonably secure IPv6 EBGP router with several of these techniques configured together. Figure 3-3 shows an example of two ISP routers that are peering with each other. Both ISPs have customer connections and their own backbone connections. The routers peer with both IPv4 and IPv6.

**Figure 3-3** *IPv6 EBGP Peering Session*



Example 3-5 shows the configuration of ISP1's R1 in this scenario. Router R1 peers with R2 over its Serial 1/0 interface. Each BGP speaker expects the TTL value in the IPv6 header to be 254. The multiprotocol BGP configuration uses the TTL-Hack and uses different passwords for the IPv4 peer and the IPv6 peer. R1 connects to the Customer 1 router over its Serial 1/1 interface. R1 uses prefix filters to limit what it learns from the customer network and what it sends and receives from the other ISP. The goal of the customer prefix list is to only allow the customer to advertise its own /48. The ISP prefix lists restrict routes more specific than a /48 and permits Teredo and 6to4 routes. Teredo and 6to4 are IPv6 transition mechanisms that are covered in more detail in Chapter 10, "Securing the Transition Mechanisms."

**Example 3-5** *Sample EBGP Router Configuration*

```
hostname R1
!
interface Loopback0
 ip address 1.1.1.1 255.255.255.255
 ipv6 address 2001:DB8::1:1:1:1/128
!
interface FastEthernet0/0
 ip address 2.2.2.1 255.255.255.0
 ipv6 address 2001:DB8:100::1/64
!
interface Serial1/0
 description ISP interconnect
 ip address 192.168.12.1 255.255.255.0
 ip access-group 100 in
 ipv6 address 2001:DB8:12::1/64
 ipv6 traffic-filter ALLOWBGP in
!
interface Serial1/1
 description Customer 1
 ip address 1.1.0.1 255.255.255.0
```

*continues*

**Example 3-5** *Sample EBGP Router Configuration (Continued)*

```
 ipv6 address 2001:DB8:1:1::1/64
!
router bgp 100
 bgp router-id 1.1.1.1
 no bgp fast-external-fallover
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 bgp maxas-limit 50
 neighbor 1.1.0.11 remote-as 1000
 neighbor 1.1.0.11 ttl-security hops 1
 neighbor 1.1.0.11 password cisco321
 neighbor 2001:DB8:1:1::11 remote-as 1000
 neighbor 2001:DB8:1:1::11 ttl-security hops 1
 neighbor 2001:DB8:1:1::11 password cisco123
 neighbor 2001:DB8:12::2 remote-as 200
 neighbor 2001:DB8:12::2 ttl-security hops 1
 neighbor 2001:DB8:12::2 password cisco123
 neighbor 192.168.12.2 remote-as 200
 neighbor 192.168.12.2 ttl-security hops 1
 neighbor 192.168.12.2 password cisco321
 !
 address-family ipv4
  neighbor 1.1.0.11 activate
  neighbor 1.1.0.11 maximum-prefix 250000
  no neighbor 2001:DB8:1:1::11 activate
  no neighbor 2001:DB8:12::2 activate
  neighbor 192.168.12.2 activate
  neighbor 192.168.12.2 maximum-prefix 250000
  no auto-summary
  no synchronization
  network 1.1.0.0 mask 255.255.255.0
 exit-address-family
 !
 address-family ipv6
  neighbor 2001:DB8:1:1::11 activate
  neighbor 2001:DB8:1:1::11 remove-private-as
  neighbor 2001:DB8:1:1::11 prefix-list FILTERV6CUSTIN in
  neighbor 2001:DB8:1:1::11 maximum-prefix 250000
  neighbor 2001:DB8:12::2 activate
  neighbor 2001:DB8:12::2 remove-private-as
  neighbor 2001:DB8:12::2 prefix-list FILTERV6ISPIN in
  neighbor 2001:DB8:12::2 prefix-list FILTERV6ISPOUT out
  neighbor 2001:DB8:12::2 maximum-prefix 250000
  network 2001:DB8:1::/48
  network 2001:DB8:1:1::/64
  no synchronization
 exit-address-family
!
access-list 100 permit tcp host 192.168.12.2 host 192.168.12.1 eq bgp
access-list 100 deny   tcp any any eq bgp
```

**Example 3-5** *Sample EBGP Router Configuration (Continued)*

```
access-list 100 permit ip any any
!
ipv6 route 2001:DB8:1::/48 Null0
!
ipv6 prefix-list FILTERV6CUSTIN seq 10 permit 2001:DB8:11::/48
ipv6 prefix-list FILTERV6CUSTIN seq 20 deny ::/0 le 128
!
ipv6 prefix-list FILTERV6ISPIN seq 10 deny 2001:DB8:1::/48
ipv6 prefix-list FILTERV6ISPIN seq 20 permit 2001:DB8::/32 le 64
ipv6 prefix-list FILTERV6ISPIN seq 30 permit 2002::/16
ipv6 prefix-list FILTERV6ISPIN seq 40 permit 2001::/32
ipv6 prefix-list FILTERV6ISPIN seq 50 deny ::/0 le 128
!
ipv6 prefix-list FILTERV6ISPOUT seq 10 deny 2001:DB8::/32 ge 49
ipv6 prefix-list FILTERV6ISPOUT seq 20 permit ::/0 le 128
!
ipv6 access-list ALLOWBGP
 permit tcp host 2001:DB8:12::2 host 2001:DB8:12::1 eq bgp
 deny tcp any any eq bgp
 permit ipv6 any any
```

## Filtering on the Peering Interface

It is a best practice to perform filtering on the interface that is used to form a BGP peering relationship. In addition to permitting transit IPv6 traffic, you should permit the BGP (TCP port 179) packets that are sourced from the directly connected BGP neighbor's address. As shown earlier in Example 3-5, both routers use ACLs to permit TCP port 179 peers from only those addresses desired. The Serial 1/0 interface has an IPv4 access list and IPv6 traffic filter that permit only BGP communications with the peer R2.

**NOTE** ISPs are also relying on another technique called infrastructure ACL (iACL). iACLs are deployed at the edge of an administrative domain and are simple ACLs that prevent the outside world from sending any packets destined to any router addresses (being loopback or physical). The only permit entries in an iACL are for BGP peering. Depending on the addressing scheme for the loopbacks and the internal links of the ISP network, these iACLs can be short and easy to deploy and to maintain.

## Using Link-Local Peering

You have already seen a secure BGP peering configuration using unicast addresses in Example 3-5. You can also configure BGP peers to use link-local addresses, but there are both benefits and drawbacks. The concept of link-local peering involves using the link-local address of the directly connected neighbor router as the IPv6 address configured for the

BGP neighbor. The concept is that if link-local addresses are used, there would be no way for any other attacker to try to create a peering session with the routers. The attacker could not communicate with either peer in the first place. Furthermore, the attacker would not know the IPv6 addresses of either peer and, as shown in Chapter 2, the reconnaissance of these addresses would not be feasible. Because many organizations might question whether to use global addresses or link-local addresses for BGP peering, it is important to cover this in more detail. The following sections review the positive and negative aspects of using link-local addresses instead of global addresses.

When using link-local addresses for BGP peers, you must explicitly configure the link-local address of the neighbor. Because DNS is not used for link-local addresses, you must manually enter these addresses. As a result, you could easily make a mistake that might take some time to troubleshoot.

Also be aware that the link-local address of a router can be shared among multiple interfaces. Therefore, you must configure the router for the neighbor's link-local address and specify the interface that is being used for the directly connected addresses. There are two ways of doing this. In earlier software versions, you would specify the interface identifier following the link-local address (for example, FE80::C800:17FF:FE88:0%Serial1/0). Another newer technique uses the **update-source** neighbor parameter to specify the interface. Example 3-6 shows how this configuration can appear.

**Example 3-6** *BGP Peering Using Link-Local Addresses*

```
hostname R1
!
interface Serial1/0
 description ISP interconnect
 ipv6 address 2001:DB8:12::1/64
 ipv6 traffic-filter ALLOWBGP in
!
router bgp 100
 bgp router-id 1.1.1.1
 neighbor FE80::C801:15FF:FE44:0 remote-as 200
 neighbor FE80::C801:15FF:FE44:0 ttl-security hops 1
 neighbor FE80::C801:15FF:FE44:0 password cisco123
 neighbor FE80::C801:15FF:FE44:0 update-source Serial1/0
 !
 address-family ipv4
  no neighbor FE80::C801:15FF:FE44:0 activate
 exit-address-family
 !
 address-family ipv6
  neighbor FE80::C801:15FF:FE44:0 activate
  neighbor FE80::C801:15FF:FE44:0 prefix-list FILTERV6ISPIN in
  neighbor FE80::C801:15FF:FE44:0 prefix-list FILTERV6ISPOUT out
  neighbor FE80::C801:15FF:FE44:0 route-map SETNEXTHOP out
  neighbor FE80::C801:15FF:FE44:0 maximum-prefix 250000
  network 2001:DB8:1::/48
```

**Example 3-6** *BGP Peering Using Link-Local Addresses (Continued)*

```
  network 2001:DB8:1:1::/64
  no synchronization
 exit-address-family
!
route-map SETNEXTHOP permit 10
 set ipv6 next-hop 2001:DB8:12::1
!
ipv6 access-list ALLOWBGP
 permit tcp host FE80::C801:15FF:FE44:0 host FE80::C800:15FF:FE44:0 eq bgp
 deny tcp any any eq bgp
 permit ipv6 any any
```

In Example 3-6, the EBGP neighbor is configured using the link-local address of the peer. The traffic filter ALLOWBGP permits communication between the peers. The interface name/number is required to be added to link-local neighbor commands because the link-local addresses are not necessarily unique to each router interface. This example uses the **update-source** method of configuring the interface for the peering session. The interface that is used is the physical serial interface that the two routers share. You should not use the loopback's link-local address as the update source when using link-local peering. This can cause confusion when troubleshooting because many of a router's interfaces share the same link-local address.

**NOTE** You can find out the link-local addresses of the routers with either the **show interface serial 1/0** command or the **show ipv6 interface brief** command.

You can also specify a link-local address that is not derived from the MAC address with the **ipv6 address ... link-local** command.

## Link-Local Addresses and the BGP Next-Hop Address

Another consideration is how BGP routers use the link-local addresses as the next hop address. A good description of how this is done is contained in the RFC 2545, "Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing." Section 3 of this RFC states that a global IPv6 address should be used as the next-hop address even though the peer can be configured to use a link-local address. This is important to consider because link-local addresses could be used on any interface and are not deterministic on which interface should be used for the communications. Because link-local addresses are local only to that subnet, they can be used across multiple interfaces without issue. However, for BGP routing, there needs to be a valid global IPv6 address that can be used for the BGP next-hop verification process.

There are situations where the next-hop attribute (MP_REACH_NLRI) can contain a single global IPv6 address or both a global address and a link-local address. The latter occurs when the two BGP peers share a common subnet, which is typically the case in EBGP. However, for IBGP, peers that might not share interfaces on a common subnet should use a global IPv6 address for their next-hop attribute.

For these reasons, a route map is required to set the next-hop address as a global address so that other routers can reach the next hop and keep this route valid. If the route map is not configured, the router will advertise one of its own global addresses as the next-hop address. If this is not reachable by the peer, the routes will be invalid and will be dropped. Most ISPs set the next hop manually to help speed convergence, so this should be an easy practice to maintain. Example 3-6 shows the configuration of the route map to explicitly set the next address.

Example 3-7 shows what the routes look like on the other EBGP router R2. The IPv6 routing table shows the route learned from R1 and the interface that the route came across on. You can also see the next-hop address in the BGP IPv6 unicast table. When you look explicitly at the route, you see the peer router's global and link-local addresses.

**Example 3-7** *Next-Hop Address for Link-Local Peers*

```
R2# show ipv6 route
IPv6 Routing Table - 12 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route, M - MIPv6
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
       D - EIGRP, EX - EIGRP external
LC  2001:DB8::2:2:2:2/128 [0/0]
     via ::, Loopback0
B   2001:DB8:1::/48 [20/0]
     via FE80::C800:15FF:FE44:0, Serial1/0
S   2001:DB8:2::/48 [1/0]
     via ::, Null0
C   2001:DB8:2:2::/64 [0/0]
     via ::, Serial1/1
L   2001:DB8:2:2::1/128 [0/0]
     via ::, Serial1/1
B   2001:DB8:11::/48 [20/0]
     via FE80::C800:15FF:FE44:0, Serial1/0
C   2001:DB8:12::/64 [0/0]
     via ::, Serial1/0
L   2001:DB8:12::2/128 [0/0]
     via ::, Serial1/0
B   2001:DB8:22::/48 [20/0]
     via 2001:DB8:2:2::22
C   2001:DB8:100::/64 [0/0]
     via ::, FastEthernet0/0
L   2001:DB8:100::2/128 [0/0]
```

**Example 3-7** *Next-Hop Address for Link-Local Peers (Continued)*

```
     via ::, FastEthernet0/0
L   FF00::/8 [0/0]
     via ::, Null0
R2# show bgp ipv6 unicast
BGP table version is 6, local router ID is 1.1.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 2001:DB8:1::/48  2001:DB8:12::1           0             0 100 i
*> 2001:DB8:2::/48  ::                       0         32768 i
*> 2001:DB8:2:2::/64
                    ::                       0         32768 i
*> 2001:DB8:11::/48 2001:DB8:12::1                         0 100 1000 i
*> 2001:DB8:22::/48 2001:DB8:2:2::22
                                             0             0 2000 i
R2# show bgp ipv6 unicast 2001:db8:11::/48
BGP routing table entry for 2001:DB8:11::/48, version 2
Paths: (1 available, best #1, table Global-IPv6-Table)
  Advertised to update-groups:
        2
  100 1000
    2001:DB8:12::1 (FE80::C800:15FF:FE44:0) from FE80::C800:15FF:FE44:0 (1.1.1.1
)
      Origin IGP, localpref 100, valid, external, best
R2#
```

## Drawbacks of Using Link-Local Addresses

As you can see, there are several security benefits of using link-local addresses for BGP peering. However, there are also some drawbacks. It is important to have identical configurations on both BGP peers, and if a change is made on one peer, the peering session can fail, causing routes to flap. If the global address changes on the interface of the EBGP peer, the BGP configuration of the EBGP peer also needs to change. As mentioned previously, BGP can carry both the link-local and global addresses in updates, so if two BGP peers share a common subnet, the MP_REACH_NLRI attribute contains both the link-local and global address. The global address is used to readvertise to other peers so that the next-hop test passes.

If the hardware changes on either BGP peer router, the corresponding addresses used in the configuration must change. The MAC address of the router's interface would be different, and the link-local address is derived partly from the MAC address. This could be a latent problem that could be difficult to troubleshoot, and it would take a small amount of effort to correct. Ironing out the details of exactly what IPv6 addresses are to be used for the BGP peer should be performed during the turn-up and provisioning procedures and also as part of the procedures for hardware replacement because of an upgrade or a failure.

It can be common practice to filter link-local addresses at the network's perimeter because link-local addresses should not be used as either the source or destination address for Internet traffic. However, filtering these packets could adversely affect EBGP, depending on how it is configured. Because there are plenty of global addresses, there is no need for peering using link-local addresses to conserve addresses.

The use of global addresses for peering keeps the configuration pretty simple. The next-hop address is simplified and global addresses are required for IBGP and EBGP multihop. There is a consistency of configuration if global addresses are used. Access lists should be used to filter BGP speakers, BTSH (TTL Hack) should be used to check the TTL value in the IP header, and the TCP MD5 signature option should be enabled. These techniques will mitigate the risk of spoofed BGP packets affecting the peering session. Therefore, these techniques can achieve the same security that using link-local addresses for peering provides.

For many, the use of link-local addresses can be overly complex. Therefore, many organizations might prefer to use global unicast addresses for EBGP peering rather than link-local addresses. Depending on your preferences, the additional work to use link-local addresses might not yield sufficient security to make it worthwhile.

## Preventing Long AS Paths

Another technique that an attacker might use against BGP is to create updates that contain unusually long AS paths. These falsified updates could put a burden on the router receiving such an update. It is not typical to have an AS path that is longer than a specific size. To prevent these paths, you can use the following BGP configuration command to limit the number of AS path hops:

```
bgp maxas-limit number-of-AS-Hops
```

This command limits the number of autonomous system (AS) numbers listed in the path of a BGP message. Typically the length of the AS path should not be more than 50 hops.

On IOS XR, you can use a configuration like the one shown in Example 3-8 to limit the number of ASNs in the path.

**Example 3-8** *IOS XR BGP Policy to Limit the AS Path Length*

```
(config)# route-policy STOPLONGPATHS
(config-rpl)# if as-path length ge 50 then
(config-rpl-if)# drop
(config-rpl-if)# endif
(config-rpl)# exit
(config)# router bgp 100
(config-bgp)# neighbor 2001:db8:100:100::1
(config-bgp-nbr)# address-family ipv6 unicast
(config-bgp-nbr-af)# route-policy STOPLONGPATHS in
```

## Limiting the Number of Prefixes Received

A similar type of attack would involve sending an extremely large number of prefixes to a peer in an effort to consume excessive amounts of memory and cause the BGP router harm. Thankfully, there are options that allow you to prevent this from happening. The following command limits the number of prefixes learned from a neighbor. This command would not only restrict the number of prefixes received from a peer, but it would also shut down the BGP peering session as a defensive mechanism if the peer sends more than 250,000 prefixes. This command was also used in Example 3-5, earlier in this chapter.

```
neighbor neighbor-ipv6-address maximum-prefix 250000
```

## Preventing BGP Updates Containing Private AS Numbers

The AS numbers in the range of 64512 to 65534 have been set aside by the IANA for private use. Therefore, these private AS numbers should not be used on the Internet or within any Internet BGP update. Therefore, you should filter any bogus paths that contain a private AS number. This is difficult to achieve using the **ip as-path access-list** commands. The following command helps make the configuration simpler and works to prevent BGP updates containing private AS numbers:

```
neighbor neighbor-ipv6-address remove-private-as
```

This command can be used on EBGP peers. This command causes the BGP router to filter out any update that has only private AS numbers. However, if the update has a mix of both private and public AS numbers, the update is allowed. Furthermore, if the update contains a list of confederated AS numbers, the private AS numbers that appear after the confederation part of the AS path list will be removed.

**NOTE** The IANA list of AS numbers can be found at http://www.iana.org/assignments/as-numbers.

## Maximizing BGP Peer Availability

BGP is used as the foundation routing protocol for the Internet. Because so many organizations worldwide rely on the stability of Internet routes, attackers would want to destabilize BGP routing if possible. BGP has several techniques to help provide stability for the Internet and help prevent attacks. However, attackers might want to get around these or even use these BGP techniques against the routers themselves to cause a DoS condition. Therefore, you should maximize the availability of your BGP peers by using these techniques.

## Disabling Route-Flap Dampening

There are attacks that target the BGP connections between peers. Even if an attacker cannot falsely inject updates, he could cause a disruption between two peers. BGP route-flap dampening was defined in IETF RFC 2439 as a way to disconnect routers from the Internet if they flapped too many times over a given period. If a router had faulty hardware, it could cause many Internet routers to add/remove routes and consume resources. An attacker can use the fact that a router is using BGP route-flap dampening against itself. Even just a few flaps could cause a neighbor to be dampened and cause an even larger outage. Some organizations can elect to not use route-flap dampening because of the DoS risks. Therefore, if you are going to use route-flap dampening, you should use the recommended parameters (RFC 2439 and RIPE-229). If you want to disable route-flap dampening, use the **no bgp dampening** BGP configuration command to turn it off.

## Disabling Fast External Fallover

One BGP optimization technique involves resetting the peer if the physical link used for that peering session failed. This is an attempt to prevent the peer from remaining up if there is an alternate path that would allow the TCP port 179 connection to remain active. Even though this feature is enabled by default, the command used to enable this feature is **bgp fast-external-fallover**. Many feel that this technique is too harsh and could cause more damage than it prevents during BGP attacks. An attack could affect the directly connected link between two peers and cause the session to fail if those routers did not have another path for communicating BGP. Therefore, you might want to disable this feature with the **no bgp fast-external-fallover** command. Disabling fast fallover means that the peer waits for the hold timer to expire before resetting the peer. You can also disable this feature on an interface basis with the **ip bgp fast-external-fallover** command. By disabling this feature, the routers are more forgiving of small outages because of an attack to prevent the BGP peering session from failing and causing a reconvergence event.

## Enabling Graceful Restart and Route Refresh or Soft Reconfiguration

If the peer does fail, you should have BGP Graceful Restart configured to speed the recovery of the peer. Graceful Restart capabilities are exchanged between peers during the OPEN message exchange. If both routers support Graceful Restart and one router comes under a short-duration attack, the other router does not discard all the routes associated with the peer but waits for the peer to recover. If the peering session is reestablished quickly, no packets are lost during the failure event. Therefore BGP Graceful Restart has security and performance benefits, but both sides of the BGP peering session must support this feature. To enable this feature, you can enter the following command within the BGP configuration block:

```
bgp graceful-restart
```

## BGP Connection Resets

If the BGP peering session fails between two routers, the routes each router has for the neighbor are eliminated. BGP peering connection resets can occur as part of standard configuration maintenance or as a result of a hardware failure or even a targeted BGP attack. You should understand the ways that the BGP routers recover from a failure.

There are two types of failures that can occur between BGP peers:

- A *hard reset* is when there is a complete failure, the entire TCP session is taken down, and all the routes are removed for that peer.
- A *soft reset* is gentler, and the peer stores prefix information until the peer is restored.

RFC 2918, "Route Refresh Capability for BGP-4," adds a route refresh capability that is exchanged when the peer is formed. Routes can be dynamically updated without having to store the updates. If you are performing normal BGP maintenance and need to reset a peer, it is better to do it with a soft reset to aid in recovery. You can see whether the BGP neighbor supports the route refresh capability by looking at the output of the **show bgp ipv6 unicast neighbor** command. Example 3-9 shows the route refresh status and the Graceful Restart capability.

**Example 3-9** *Viewing the BGP Peer Status*

```
R1# show bgp ipv6 unicast neighbor
BGP neighbor is 2001:DB8:12::2,  remote AS 200, external link
  BGP version 4, remote router ID 1.1.1.2
  BGP state = Established, up for 00:02:11
  Last read 00:00:11, last write 00:00:11, hold time is 180, keepalive interval
is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received(old & new)
    Address family IPv6 Unicast: advertised and received
    Graceful Restart Capability: advertised and received
      Remote Restart timer is 120 seconds
      Address families preserved by peer:
        none
...
```

If the route refresh capability is not available on either peer, you can configure soft reconfiguration. This can be done with the following two commands:

```
bgp soft-reconfig-backup
neighbor neighbor-ipv6-address soft-reconfiguration [inbound]
```

## Logging BGP Neighbor Activity

It is also a best practice to log all BGP neighbor activity. If an attacker is targeting your BGP routers, you should log all BGP neighbor changes. This is a good practice for typical operational reasons besides the security-monitoring aspects. Following is the command that needs to be configured under the **router bgp** stanza:

```
bgp log-neighbor-changes
```

## Securing IGP

Because BGP is a TCP layer routing protocol, it relies on a stable IP foundation. In fact, BGP oftentimes relies on a stable IGP to be able to reach the next hop or a distant IBGP peer. Therefore, the security of the IGP routing protocol is important. Chapter 6 shows several configurations on how to secure various IGPs. If you are using Intermediate System–to–Intermediate System (IS-IS) as your IGP, be sure to use the optional password-protected checksums defined in RFC 3358. Within the service provider's network, use Open Shortest Path First version 3 (OSPFv3) with IPsec instead of just MD5 authentication. These practices can help prevent attackers from making your BGP architecture fail.

## Extreme Measures for Securing Communications Between BGP Peers

Other techniques for securing communications between BGP peers are outside the configuration of BGP but can help support the security of the BGP communications. Drastic measures for securing peering can include turning off the Neighbor Discovery Protocols (NDP). Because of the IPv6 risks on LANs that are similar to the risks found in IPv4 Address Resolution Protocol (ARP), you can elect to statically define the IPv6 addresses on the interfaces.

If there are no hosts on the Ethernet interface between the two BGP routers, there is little use for NDP to operate. Disabling NDP would be synonymous with using static ARP entries in an IPv4 LAN for Ethernet peering. On IPv6 networks, this means configuring static MAC addresses and binding them manually to link-local addresses, thereby creating static neighbor cache entries. This would take the guesswork out of configuration of the neighbor, and the NDP would not be required for normal operations. Furthermore, you could consider using static content-addressable memory (CAM) entries in any Ethernet switches between the BGP peers. These techniques are only for the extremely paranoid and for those network administrators with lots of time on their hands. These techniques could have additional side effects that require more configuration commands, additional troubleshooting, and higher operational costs that do not justify a small gain in additional security.

# IPv6 over MPLS Security

When service providers consider their IPv6 deployment plans, they look at what services their customers want and what the customers are willing to pay for. They then consider how difficult it would be to provide these services with the infrastructure they already have. Because service providers might not be charging extra for IPv6 connectivity, the budget for the deployment is extremely low. Therefore, the simplest methods of deploying IPv6 are often preferred.

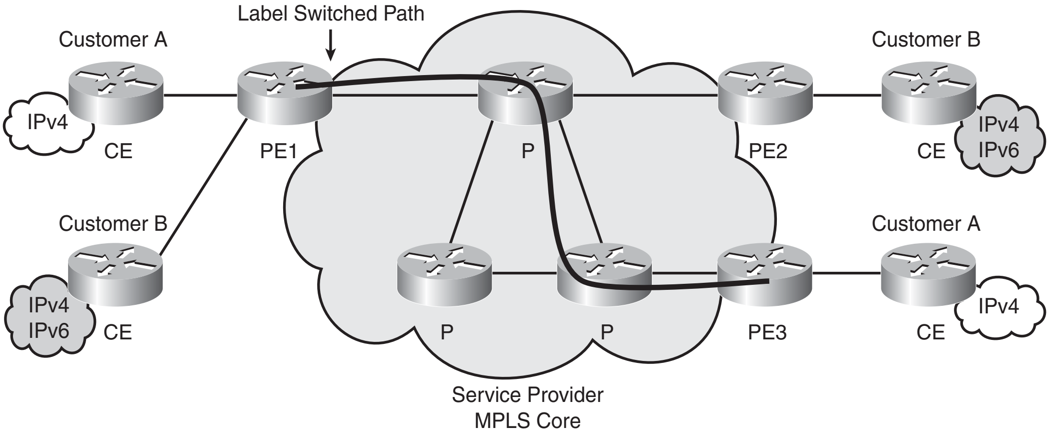
RFC 4029, "Scenarios and Analysis for Introducing IPv6 into ISP Networks," describes the steps of IPv6 deployment that most service providers take into consideration. Service providers start by creating a dual-stack backbone and connecting to an IPv6 exchange. Service providers initially create connections using tunnels. As their migration progresses, customers can be connected with native dual-stack connections. This involves the use of an IPv6-capable IGP such as IS-IS or OSPFv3. Eventually their entire infrastructure is fully dual-stack capable. However, this takes considerable time and can require investment in new equipment that is dual-stack capable.

Some service providers use IPv6 tunnels over their existing IPv4 infrastructure to provide IPv6 services to their customers. They find that despite the scalability issues of maintaining multiple manually configured tunnels, it is easy to configure. The downsides are that troubleshooting is more difficult because IPv6 connectivity is based on the underlying IPv4 network stability. Tunnels can also route traffic in awkward ways that can be suboptimal and increase latency. The other concern is that ultimately these tunnels will have to be taken down as the network becomes fully dual-stack capable.

Many service providers have already deployed IPv4 Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPN) (RFC 4364). Figure 3-4 shows a service provider's MPLS network that supports customers that use different IP versions. Customers use customer edge (CE) routers at their sites to communicate with the service provider's provider edge (PE) router. PE routers use Virtual Routing and Forwarding (VRF) instances to separate customers into their own VPNs. Inside the service provider's core provider (P), routers create label switched paths (LSP) to connect customer sites but prevent customers' networks from communicating with other customers. As customer packets traverse the MPLS, core network labels are used at each hop to help forward the packets. MPLS networks can be leveraged for providing IPv6 services to customers. There are several models of adding IPv6 to an existing IPv4 MPLS network, including the following:

- Use static IPv6 over IPv4 tunnels between PE routers
- Use 6PE (simpler PE routers that are IPv6-aware) to use the IPv4 MPLS core to send IPv6 packets between PE routers
- Use 6VPE (MPLS VPN dual-protocol PE routers) to create separate IPv6-aware VRFs

**Figure 3-4** *Dual-Protocol MPLS VPN*



Service providers can offer several types of IPv6 services. Every flavor has its advantages and disadvantages for the service provider and the customer. You should know which one you are purchasing from the service provider. This can help you determine the risks that exist and know how to mitigate them. The following sections provide an overview of each model. In-depth coverage of how to configure each of these types of networks is outside the scope of this book. However, *Deploying IPv6 Networks*, by Ciprian P. Popoviciu, Eric Levy-Abegnoli, and Patrick Grossetete (Cisco Press, 2006), covers the configuration details of setting up these different types of MPLS environments for IPv6.

## Using Static IPv6 over IPv4 Tunnels Between PE Routers

The first technique mentioned uses statically configured tunnels between PE routers. The tunnel interfaces have a tunnel destination of the remote PE router's IPv4 address. The IBGP between PE routers and the LSPs created over the IPv4 P routers allow the tunnel endpoints to communicate. The security issues related to static tunnels apply to this solution. You should make sure that you are protecting the tunnel endpoints and filtering traffic entering and leaving the tunnel at both ends. The goal is to prevent spoofed packets from entering the tunnel or escaping the tunnel.

## Using 6PE

The second technique involves enabling IPv6 on the PE routers and using the IBGP advertisements of the IPv6 routes to form LSPs that can carry the IPv6 packets. A two-label stack is typically used. The inner label is the BGP label for the IPv6 route. The outer label is based on the P routers forwarding the traffic based on their IPv4 IGP routing protocols. This technique is called 6PE.

The advantage of using 6PE is that the core can remain IPv4-only for the near term, and the LSPs are constructed between dual-stack PE routers. The signaling of the LSPs still uses IPv4. Although 6PE does not support IPv6 multicast, it is one of the easiest ways to leverage an existing IPv4 MPLS core infrastructure. This can give the service provider time to upgrade its MPLS core to IPv6 while still providing basic dual-protocol services to customers.

Similar to the tunneling methods, 6PE will eventually need to be migrated away from IPv4 as the core gets migrated to IPv6. Because 6PE is not the final solution, it is considered by some to be just an incremental step toward a fully IPv6-aware core network. The work to migrate to IPv6 is tough enough without having to go through many intermediary steps that can become migrations in and of themselves. As they say, “If you don’t have time to do it right, you certainly don’t have time to do it over.” For this reason, some aggressive service providers might bypass the 6PE step and strive for an IPv6-aware core directly.

Another disadvantage of the 6PE technique is that 6PE is like having one large single routing table. There is no differentiation of customer traffic across the core. Customers are not separated from each other as with Layer 3 MPLS-based VPNs. 6PE is more like having a big MPLS Internet service with global IPv6 routes. This technique can be used for commodity IPv6 Internet connectivity for customers. If you are using an MPLS service for Internet connectivity, you need to protect your perimeter accordingly. If you are using a 6PE service for site-to-site connectivity, you should be filtering traffic going between sites and filtering the routes being advertised and received from the service provider. You might also want to consider using encryption between your sites as an extra measure of security.

The security implication of using 6PE services is that there is no inherent security built into the service. Customers should be aware of the type of service they are selecting from the provider and protect their traffic accordingly. Just because the service provider says that the IPv6 is being provided over an MPLS network, do not assume that a Layer 3 MPLS-based VPN service is being used.

## Using 6VPE to Create IPv6-Aware VRFs

The third solution is an IPv6 MPLS VPN service. 6VPE is more like the MPLS-based VPNs that are currently popular for IPv4 connectivity. Using a Layer 3 MPLS VPN service for IPv6 networks gives the security benefits of separating customer traffic into different VRFs. 6VPE networks use a two-label stack, with the internal label being the VPN label identifier and the outer label being assigned as a result of the IGP. The P routers only look at the label and swap labels. They do not care whether it is an IPv4 or IPv6 packet inside. At the same time, 6VPE should fit the operational models that many service providers have already adopted. However, there currently are limited solutions for creating native IPv6 LSPs using Label Distribution Protocol (LDP) or Resource Reservation Protocol (RSVP). While LDPv6 has been defined, it is not widely implemented. The Cisco 6VPE solution is an implementation of RFC 4659. 6VPE can work on top of a core infrastructure that uses either or both IP versions. That can mean infrastructure upgrades and the deployment of an IGP that is capable of both IPv4 and IPv6 routing.

6VPE provides the same level of security as IPv4 BGP-based Layer 3 MPLS VPNs, which is discussed in RFC 4381, "Analysis of the Security of BGP/MPLS IP Virtual Private Networks (VPNs)." As long as the 6VPE PE routers are configured properly, the system can provide the same security as traditional ATM or Frame Relay links. Because the service provider isolates its core network from customers, malicious customer traffic cannot impact the control plane of the service provider's out-of-band management network. Therefore, 6VPE can be safer than many other forms of WAN services.

**NOTE** Another book analyzes in detail the security of MPLS networks: *MPLS VPN Security*, by Michael H. Behringer and Monique J. Morrow (Cisco Press, 2005). Most of this book is also applicable to 6VPE services.

## Customer Premises Equipment

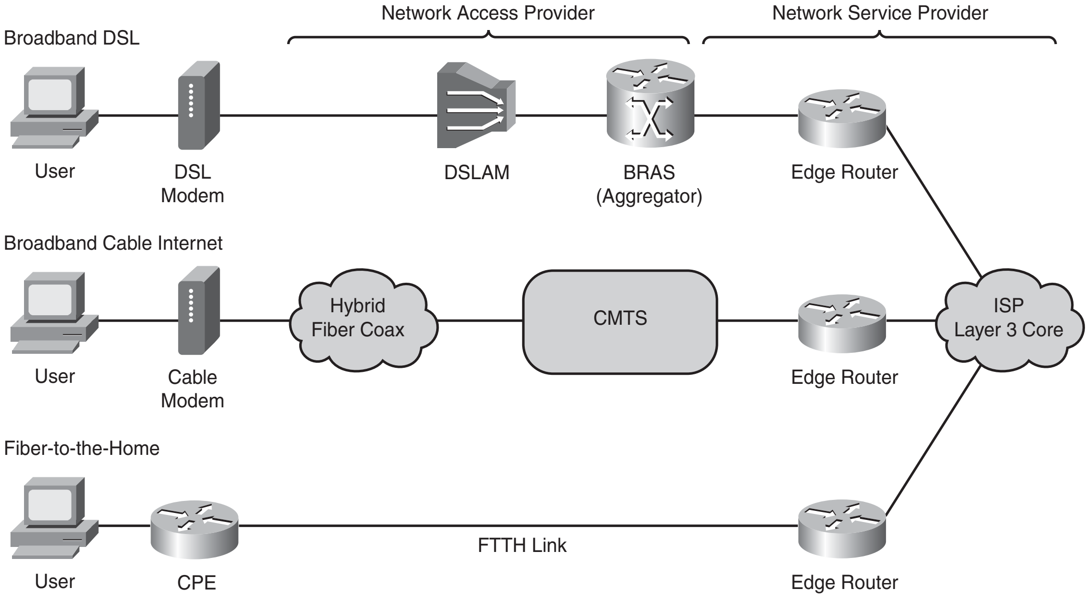
Service provider networks need to connect to many orders of magnitude more remote devices than enterprise networks do. Whereas a typical enterprise might have fewer than a hundred remote sites, a service provider could have thousands if not millions of subscribers. That means that scalability is of the utmost importance, and the reliability of the network must be maintained. IPv6 can uniquely support the addressing requirements for these types of networks.

No matter what type of physical medium the customer connection uses (xDSL, Cable/HFC, Fiber to the Home [FTTH], wireless), networking equipment called customer premises equipment (CPE) terminates service at the customer site. This equipment terminates the type of service provided by the Network Access Provider (NAP) and contains the Layer 3 address provided by the Network Service Provider (NSP). Each type of broadband access has its own way of connecting the customer to the NSP's routed infrastructure. Broadband access connections can use a direct connection or some form of tunneled protocol such as PPP or L2TP to connect the CPE to the Internet.

The service provider must secure its own network infrastructure when providing IPv6 services. Figure 3-5 shows several broadband access provider network topologies.

Regardless of whether DSL Access Multiplexer (DSLAM)/Broadband Remote Access Server (BRAS) or Cable Modem Termination System (CMTS) devices are used, the edge router (ER) is the device that connects the customer's connection to the IPv6 Internet. These devices, and particularly the ER, must be hardened from a security perspective. The NAP and NSP should be able to keep track of which user's CPE has been allocated to which address or address block. This is typically done with RADIUS servers that authenticate the user's connection before allowing them on the network.

**Figure 3-5** *Broadband-Access Provider Topologies*



CPE for small offices/home offices (SOHO) or residential broadband access needs to be easy to configure, administer, and secure. This is a requirement because the users are not necessarily knowledgeable about IPv6 or even networking. If the device has advanced settings that give more control over the IPv6 connectivity and security, the default settings should be set to make it easy and yet secure. Otherwise, the service provider can have substantial support calls from customers to help them configure their CPE devices. Even though the end user should be concerned about the security of his connection to the Internet, he does not necessarily need to be bogged down in the details. Service providers must consider the customer's security and their end-user experience when selecting CPE devices on their behalf.

These home-user-grade products are the simplest form of routers. They have a single default route, and they provide DHCP services to the computers on their wired or wireless LAN. They gain a single public IP address and perform Port Address Translation (PAT) for the private addresses used behind the public IP. The security functions they provide involve simply preventing inbound connections from being made. Simple dual-stack residential devices have similar features for both protocols. However, for the IPv6 protocol, PAT is not necessary because global unicast addresses are used for the CPE device's external and internal interfaces.

Residential and SOHO security devices should perform some of the same functions as commercial-grade firewall products. Residential gateways need to be able to statefully permit outbound connections and only allow inbound packets that result from an outbound connection. If a CPE product does allow more advanced configuration of the firewall policy, the default settings should be used to prevent Internet traffic from reaching the internal LAN.

Consumer-grade CPE best practices include the following:

- Do not forward packets that have a multicast source address.
- Block packets destined for multicast destinations in the outbound direction.
- Do not allow RH0 packets inbound or outbound.
- Block packets sourced from Unique Local Address (ULA) space (that is, FC00::/7).
- Block other bogon addresses from entering and leaving the interior LAN (difficult to do because this list changes several times each year).
- Block packets that are not sourced from the global unicast prefix assigned on the LAN interface. This prevents spoofed packets from leaving the user LAN.
- Deny packets sourced from the internal LAN prefix from coming in the external interface. This prevents spoofed packets from entering from the Internet.

IPv6-capable CPE routers should also prevent Teredo tunnels from forming from internal clients to Teredo servers. Teredo is only a transition mechanism for IPv6-capable hosts behind IPv4 NATs. Teredo is not used if the client has native IPv6 connectivity through the CPE router to an IPv6-capable service provider. The risk is that Teredo tunnels can be used as a back door into the client computer. However, preventing Teredo tunnels from being established can be difficult to accomplish. More information on this subject appears in Chapter 10.

If consumer-grade CPE devices are constructed with IPv6 security measures enabled by default, the customer's Internet connection will be more secure. The end user does not have to be so worried about these details, and the device can provide the required security features right out of the box.

More advanced users, like those reading this book, might want to have a more sophisticated device at their homes. For more advanced users that require more power, CPE devices such as the Cisco ASA5505, the Cisco 871, or the newer Cisco 880/860 routers perform nicely. These devices have full IPv6 capabilities and the ability to filter IPv6 packets based on a wide variety of header fields and extension headers. These devices might not have these default security settings. Therefore, the more advanced users need to be able to configure these same settings to secure their own Internet connections.

**The Delicate Balance Between a Secure CPE and an Open CPE**

There are heated discussions between the proponents of a secure CPE (like the one described in this book) and those of a more open CPE, which would allow any incoming IPv6 connections. The latter CPE makes several legal peer-to-peer applications possible (like voice or any other collaboration system) because any IPv6 node can then connect to any IPv6 node. This was impossible to achieve in the IPv4 world because of lack of IPv4 addresses, but it is possible in the IPv6 world.

The default security policy on residential CPEs is expected to accommodate multiple security zones—some with a relaxed policy (for video collaboration) and others with a strict policy (for the usual computers or for the video surveillance network) .

## Prefix Delegation Threats

Service providers need to connect numerous customers to the IPv6 Internet. Most ISPs will connect larger customers with dedicated interfaces. These could either be T1s, Metro Ethernet, fiber, SONET, wireless, or any of a variety of media types. These directly connected customers will receive address assignments from the allocations that the service provider received from the regional registry. These assignments are performed manually and require coordination between the customer and the service provider. This method of allocating addresses is possible but does require the customer to be savvy at configuring his CPE.

For service providers that must connect millions of IPv6 Internet subscribers, there is no feasible way to coordinate direct assignments to that many customers manually. There needs to be an automated way of allocating IPv6 prefixes to customers and reclaiming those assignments if the customer disconnects. Current IPv4 broadband providers give customers a single IPv4 address and let the customer's device perform NAT. IPv6 will allow customers to acquire much more public address space. Broadband customers could be allocated a /48, /56, or /64 network prefix depending on the provider's policies, and then their CPE would allow the customers' hosts to perform Stateless Address Auto-configuration (SLAAC).

The following sections describe the use of SLAAC and indicate why some service providers prefer to use DHCPv6 instead.

## SLAAC

Provisioning of new customer connections must be automated in some way to have a scalable system for the broadband service providers to maintain. One technique is to leverage SLAAC to allow the CPE device or hosts to acquire public IPv6 addresses. SLAAC can be used to uniquely allocate the addresses, and the Neighbor Discovery Protocol (NDP) function Duplicate Address Detection (DAD) can be used to avoid addressing conflicts. SLAAC might not be the best option for allocating IPv6 addresses to customers because there are no security features within the NDP. Furthermore, SLAAC can be a simple way to have nodes determine their address, but it does not provide them with other necessary information for communications, such as a DNS server for the node to use.

## DHCPv6

Because SLAAC does not do everything that a service provider wants, the provider can elect to use DHCP version 6 (DHCPv6). The service provider's Layer 3 edge router can send a router advertisement (RA) message to inform customers that DHCPv6 is in use. The RA sends the A/M/O bits to tell the node that DHCPv6 is available. There can still be concern that the RA messages could be spoofed by an attacker. Because of the security issue of spoofed RA messages, service providers might want to make use of DHCPv6 instead of SLAAC. That way, they can know exactly who is turning up on the network.

Service providers might want some type of authentication to take place to verify a customer's legitimacy before allowing the customer on the network. If the subscriber has not paid his bill, he will not be allowed on the Internet. To gain more control over the subscriber, a service provider might want to use DHCPv6 rather than SLAAC. There can also be a concern that attackers could spoof DHCPv6 servers or DHCPv6 relays. Rogue DHCPv6 servers could give out false information. Therefore, the security of DHCPv6 is a serious concern.

There are some solutions to the security vulnerabilities within DHCPv6. Hackers could also try to see whether DHCPv6 servers are allocating sequential lease addresses. That would lead to much easier network reconnaissance. Cisco Network Registrar gives out pseudorandom leases, so this would prevent easy guessing of the client assigned addresses.

Another risk is that a single system could consume DHCPv6 resources similar to the way that the hacker utility Gobbler can eat up all the available IPv4 DHCP addresses. One possible solution to the resource consumption attack is to rate limit messages sent to FF02::1:2 (All DHCPv6 Relay Agents and Servers) and FF05::1:3 (All DHCPv6 Servers).

If attackers can observe the information between the client and the server, many problems would result. DHCPv6 offers a mechanism to secure communication from the client and the DHCPv6 server with the use of authentication algorithms. This authentication mechanism does not provide confidentially but merely helps prevent theft of service. Within the DHCPv6 protocol itself, there is no current way to secure communications between the DHCPv6 relay agent and server. Separate IPsec configurations could be used to secure these communications.

DHCPv6 can provide a prefix to a device in addition to providing individual IPv6 addresses to hosts on a LAN. This is an extension to the DHCPv6 specification called DHCPv6 Prefix Delegation (DHCPv6-PD). The client device acts as a DHCPv6 client, and the DHCPv6 delegating router acts like the DHCPv6 server. It is relatively simple to have one router be a DHCP server for other access routers. The delegating router can be preconfigured with a pool of addresses that prefixes will be allocated from. The client router configuration is equally simple.

**NOTE** *Deploying IPv6 Networks*, by Ciprian Popoviciu, Eric Levy-Abegnoli, and Patrick Grossetete (Cisco Press, 2006), offers good examples of DHCPv6-PD in Chapter 3.

Example 3-10 shows what a delegating router configuration might look like. The DHCPv6 configuration on the router is tied to a specific interface. A pool is created that defines the block of addresses to allocate from and the prefix length to give to the client. In this case, /48 blocks are delegated to the clients out of a /40 pool. A DHCPv6 pool is created and assigned to an interface.

**Example 3-10** *Delegating Router Configuration*

```
hostname R1
!
ipv6 unicast-routing
ipv6 dhcp pool CUSTPOOL
 prefix-delegation pool PREFIX
 dns-server 2001:DB8:1::1
!
interface FastEthernet1/0
 description Link to customers for DHCP prefix delegation
 no ip address
 ipv6 address 2001:DB8::1/64
 ipv6 dhcp server CUSTPOOL
!
ipv6 local pool PREFIX 2001:DB8:FF00::/40 48
```

The configuration of the DHCPv6 client is simple. Example 3-11 shows that DHCPv6-PD is tied to an interface and the allocated prefix is assigned to a general prefix variable. Router R2 is connected to R1 with interface Fast Ethernet 1/0. This general prefix variable can be used on other downstream interfaces.

**Example 3-11** *Client Router Configuration*

```
hostname R2
!
interface FastEthernet1/0
 description Link to ISP for DHCP prefix delegation
 no ip address
 ipv6 address autoconfig default
 ipv6 enable
 ipv6 dhcp client pd PREFIX
!
interface FastEthernet1/1
 description LAN Link that will inherit prefix
 no ip address
 ipv6 address PREFIX ::1:0:0:0:1/64
 no keepalive
```

After these routers are configured and the Fast Ethernet 1/0 interface comes up, the delegating router can see the DHCPv6 requests and allocate the block. Example 3-12 shows the status of the delegating router. You can see the /48 block allocated to the client and the identity of the client device.

**Example 3-12** *Delegating Router Status*

```
R1# show ipv6 local pool PREFIX
Prefix is 2001:DB8:FF00::/40 assign /48 prefix
1 entries in use, 255 available, 0 rejected
0 entries cached, 1000 maximum
User                Prefix                                       Interface
00030001CA0117DC000000050001
                    2001:DB8:FF00::/48
R1# show ipv6 dhcp bind
Client: FE80::C801:17FF:FEDC:1C
  DUID: 00030001CA0117DC0000
  Interface : FastEthernet1/0
  IA PD: IA ID 0x00050001, T1 302400, T2 483840
    Prefix: 2001:DB8:FF00::/48
            preferred lifetime 604800, valid lifetime 2592000
            expires at Sep 12 2008 08:09 AM (2590587 seconds)
R1# show ipv6 dhcp interface
FastEthernet1/0 is in server mode
  Using pool: CUSTPOOL
  Preference value: 0
  Hint from client: ignored
  Rapid-Commit: disabled
R1# show ipv6 dhcp pool
DHCPv6 pool: CUSTPOOL
  Prefix pool: PREFIX
               preferred lifetime 604800, valid lifetime 2592000
```

**Example 3-12** *Delegating Router Status (Continued)*

```
  DNS server: 2001:DB8:1::1
  Active clients: 1
R1#
```

The client router now has the allocated address assigned to its interfaces. Example 3-13 shows the status of the client router after the DHCPv6-PD allocation has been made. The **show ipv6 dhcp** command shows the client's DHCP Unique Identifier (DUID). The DUID can be unique to the client device, and DUIDs are assigned by the client router automatically and are based on the lowest MAC address on the device.

**Example 3-13** *Client Router Status*

```
R2# show ipv6 dhcp
This device's DHCPv6 unique identifier(DUID): 00030001CA0117DC0000
R2# show ipv6 dhcp interface FastEthernet 1/0
FastEthernet1/0 is in client mode
  State is OPEN
  Renew will be sent in 3d11h
  List of known servers:
    Reachable via address: FE80::C800:17FF:FEDC:1C
    DUID: 00030001CA0017DC0000
    Preference: 0
    Configuration parameters:
      IA PD: IA ID 0x00050001, T1 302400, T2 483840
        Prefix: 2001:DB8:FF00::/48
                preferred lifetime 604800, valid lifetime 2592000
                expires at Sep 12 2008 08:09 AM (2590412 seconds)
      DNS server: 2001:DB8:1::1
      Information refresh time: 0
  Prefix name: PREFIX
  Rapid-Commit: disabled
R2#
```

The DUID can be used to provide some minor form of security for the DHCPv6-PD communications. DUIDs can be assigned statically, and the DUID could be assigned by the service provider. This might be slightly more secure, but it would eliminate any efficiency gained by using an automated address assignment method. If the DUID needs to be configured manually on the CPE, DHCP-PD might not be of much benefit compared to manually assigning a block to a customer.

Example 3-14 shows how the DUID can be statically configured on the delegating router R1. In this example, the prefix is granted only to the client router R2 with the preconfigured DUID.

**Example 3-14** *Delegating Router with Static DUID*

```
ipv6 dhcp pool CUSTPOOL
 prefix-delegation 2001:DB8:1234::/48 00030001CA0117DC0000
 dns-server 2001:DB8:1::1
```

When this change is made on R1 and R2 reconnects to the service provider network, R2 receives a unique delegation based on its DUID. Example 3-15 shows the new address that R2 has been given. Because R2 is using a general prefix, it is passing along the use of that prefix to its Fast Ethernet 1/1 interface address.

**Example 3-15** *Client Router with Static DUID*

```
R2# show ipv6 dhcp interface FastEthernet 1/0
FastEthernet1/0 is in client mode
  State is OPEN
  Renew will be sent in 00:00:46
  List of known servers:
    Reachable via address: FE80::C800:17FF:FEDC:1C
    DUID: 00030001CA0017DC0000
    Preference: 0
    Configuration parameters:
      IA PD: IA ID 0x00050001, T1 60, T2 120
        Prefix: 2001:DB8:1234::/48
                preferred lifetime 604800, valid lifetime 2592000
                expires at Sep 12 2008 08:38 AM (2591987 seconds)
      DNS server: 2001:DB8:1::1
      Information refresh time: 0
  Prefix name: PREFIX
  Rapid-Commit: disabled
R2# show ipv6 interface brief
FastEthernet1/0            [up/up]
    FE80::C801:17FF:FEDC:1C
    2001:DB8::C801:17FF:FEDC:1C
FastEthernet1/1            [up/up]
    FE80::C801:17FF:FEDC:1D
    2001:DB8:1234:1::1
R2#
```

Even with statically defined DUIDs, there can still be risks to DHCP-PD that could make this type of addressing problematic. An attacker could spoof a DUID or somehow try to impersonate another customer connection. This could either cause a misdirection of traffic or cause a DoS situation for the legitimate user. The same threats against traditional DHCP are the same as the threats against DHCPv6-PD.

If you wanted to make your address allocation system more secure, you could use a RADIUS server to authenticate the prefix delegation. You could create other ways to secure the DHCPv6 messages, but that would require more preconfiguration on the customer's equipment. The purpose of DHCPv6-PD is to make addressing simpler. If more coordination and expectations are placed on the skill of the broadband subscriber, the efficiency benefits will be lost.

## Multihoming Issues

IPv6 addresses are allocated by service providers to end-user organizations. IPv6 addresses are intended to be fully hierarchical to help reduce the size of the core Internet routing table. Because IPv6 has the ability to have far more address blocks than IPv4, it would be impossible to have a large number of routes in the Internet backbone routers. With the increasing size of today's IPv4 Internet routing table, many devices struggle to handle the storage and the workload of processing the changes. Both memory and processor capacity are factors in the maximum size of the IP routing table. The size of the Forwarding Information Base (FIB) and the Routing Information Base (RIB) increases with the number of routes. As the FIB gets larger, so does the lookup time, which affects the forwarding rate. As the size of the routing table increases, so does the time of convergence. If Internet routers contain both IPv4 and IPv6, the problem gets worse.

Because IPv6 addresses are fully hierarchical, you probably do not need to use BGP, except in the default-free zone of the Internet backbone. An ISP could simply use a static route to point to the address block that has been allocated to the customer. In turn, the customer could simply use a default route to point toward the ISP for routing traffic to all unknown prefixes. This would simplify device configurations and also reduce the need for BGP, which would reduce the number of protocols the routers needed to run.

Many large organizations that connect to today's IPv4 Internet enjoy the redundancy that comes from connecting to two or more ISPs. This is part of an enterprise organization's disaster recovery and business continuity plan. The organization takes in routes from these providers (full routes, partial routes, or just the default route) and advertises its own address space from its own Autonomous System Number (ASN). Therefore, if one ISP connection were to fail, the BGP routing tables would converge and the customer would maintain its Internet connectivity.

If the rules of IPv6 addressing hierarchy were relaxed, many organizations could advertise their prefixes to the Internet. The address space would become fragmented, and the size of the Internet routing tables would expand out of control. Because of this fear, the addressing hierarchy has been enforced by the IANA, the IETF, the regional registries, and the ISPs. However, various registries (notably ARIN) have started to allow customers to obtain provider independent (PI) address space. This address space is not likely to be routed by service providers, but it does give customers additional addresses should they need them.

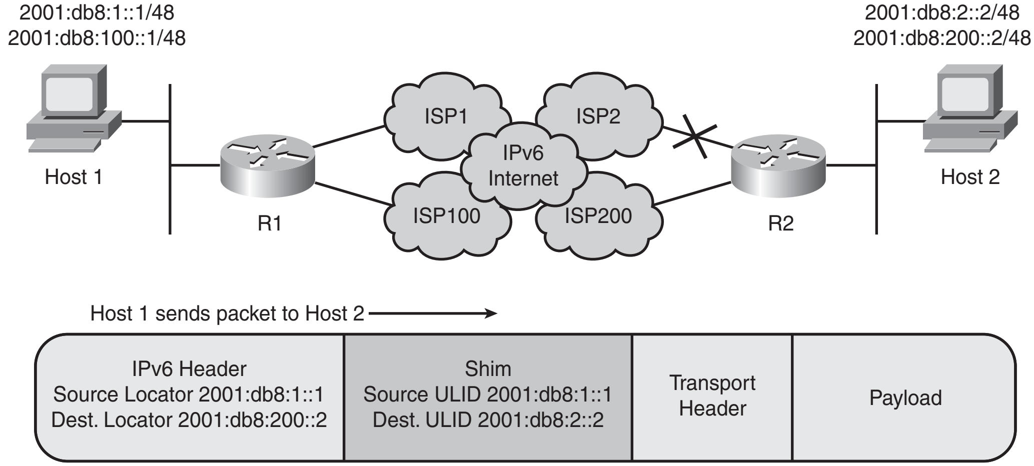
Many larger organizations still have a desire to have redundant connections to the Internet. Multinational organizations want to have Internet connections on the different continents they operate, for example. This is a requirement to reduce the latency that would result in back-hauling their Internet traffic to one central Internet attachment point. The redundancy and availability needs of customers must be addressed in some way. Customers must be allowed to be multihomed to the Internet. However, problems arise when sites have multiple address assignments from multiple ISPs. If one ISP link goes down, the other ISP does not readvertise the other ISP's address space. The customer addresses its web servers in one ISP's address space, and if that ISP fails, the web servers cannot be reached through the other ISP link. Therefore, alternatives must exist to allow the redundancy and failover between service providers without violating the address hierarchy rule.

The IETF has performed much work on the subject of multihoming. This early work is documented in RFC 3582, "Goals for IPv6 Site-Multihoming Architectures." Now the Site Multihoming by IPv6 Intermediation working group (shim6) is developing solutions to address sites that are multihomed. The primary solution that exists today is to use a ”shim“ that can be a new layer between the network layer and the transport layer. Above the shim are stable routable IPv6 addresses that allow applications to work as they have before and do not disrupt DNS information. The addresses above the shim are called Upper Layer IDs (ULID). Below the shim, IPv6 addresses can be used from either assigned blocks to get the packets forwarded to the destination.

Two hosts that want to communicate reliably both need to support the shim layer, and an initial shim protocol exchange needs to take place. During this exchange, both shim hosts share their available addresses with each other. This exchange shares the locator IDs between the two hosts. After this protocol exchange, both hosts are communicating with each other. If one of the address blocks loses connectivity because of an ISP failure, it can simply switch to using the other address space.

Figure 3-6 shows an example of how shim6 might work. Two sites have connections to two ISPs each, and each site has been allocated two /48 prefixes each. The two hosts need to communicate with each other, regardless of which ISP is available. They first communicate over whichever address space is available and then perform their shim protocol exchange. During this exchange, they share with each other their list of locator IDs, which are the address blocks the sites have been assigned by their ISPs. They are then able to communicate by using the shim header that contains the ULIDs. If host 2 loses its ISP2 connection, host 1 can use the locator ID for the remaining available prefix for host 2 that is still operational. Notice that the ULIDs did not change and thus the applications maintained state.

**Figure 3-6** *Shim6*



If an attacker could spoof packets with the shim header, several types of vulnerabilities would exist. One possible set of attacks comes from an attacker that is in the middle of the communication between two shim6 hosts. That attacker could perform redirection attacks to try to hijack the session. If the attacker could impersonate the locator IDs and the ULIDs, he could take over the communications. If the attacker could get a host to cache a locator ID, the attacker could redirect traffic to another network for an extended period of time.

Another type of attack would be a flooding attack, where an attacker would use its own locator ID to redirect a large volume of traffic to the victim. However, shim6 hosts perform a reachability probe-and-reply process to determine that the locator ID belongs to the remote host.

One solution to these security issues is for both hosts to use Hash Based Addresses (HBA) to ensure authenticity of the two hosts' locator IDs. These HBAs are a cryptographic one-way hash of the set of prefixes available for communications. This provides hijack protection because the HBAs cannot be tampered with in transit without detection. Performing the hash using nonces also helps prevent against replay attacks. Some form of public-key infrastructure (PKI) mechanism could also be used to secure the exchange between hosts.

There are additional security implications of using a shim between the IPv6 header and the upper-layer headers. Firewalls need to keep track of multiple sets of address space from different providers. This means that the firewall policies will grow, and the complexity of maintaining the rules and the management overhead will also grow. This is because hosts will have multiple addresses that could be used to source packets that can make it difficult to create granular firewall policies. Firewalls need to be shim-aware and parse the packets carefully, and they need to be able to handle sessions that start out without a shim and then transition to using a shim. Packet filters also need to be aware of session state when the ULIDs change within the shim.

Currently discussions are ongoing within the IETF about the use of shim6 and how it impacts other aspects of the IPv6 protocol and the operations of an IPv6 network. There are only a couple of implementations for hosts. There are discussions about integrating this functionality into routers so that they can perform this process on behalf of devices that do not have sufficient resources to create the shim themselves. There is also discussion about how the shim could be used for traffic engineering purposes instead of a simple multihoming solution. For the most updated information on this topic, you can go to the shim6 IETF working group site at http://www.ietf.org/html.charters/shim6-charter.html.

# Summary

There will be many large-scale Internet threats that plague the IPv6 Internet in just the same way as DoS attacks disrupt today's Internet. Hopefully the larger address space of IPv6 will make scanning worms a thing of the past; however, other types of worms are likely to evolve. If service providers and customer organizations are performing ingress and egress filtering, tracebacks will be easier. The more research done on these IPv6 Internet threats, the more secure the IPv6 Internet will be in the future.

Service providers might be hesitant to add IPv6 functionality to their production IPv4 networks. They have a fear that new IPv6 vulnerabilities will lead to instability of their revenue-generating IPv4 networks. Network service providers can leverage secure BGP peering to help make the Internet a safer place for all. If service providers perform the proper filtering, they can mitigate many of these risks. Many organizations are connecting to dual-stack services today, and service providers can leverage their existing MPLS infrastructures to create secure IPv6 services.

The key is to make the customers' experience transparent, which means making it easy for them to configure their devices and securely automate address assignments. However, customers will have the same demands of IPv6 Internet connectivity as they have with IPv4 Internet connectivity. That means that solutions to IPv6 multihoming will need to be developed and secured.

# References

AfriNIC. ftp://ftp.afrinic.net/pub/stats/afrinic/delegated-afrinic-latest.

APNIC. http://ftp.apnic.net/stats/apnic/delegated-apnic-latest.

ARIN. ftp://ftp.arin.net/pub/stats/arin/delegated-arin-latest.

Asadullah, S., A. Ahmed, C. Popoviciu, P. Savola, and J. Palet. RFC 4779, "ISP IPv6 Deployment Scenarios in Broadband Access Networks." http://tools.ietf.org/html/rfc4779. January 2007.

Baker, F. and P. Savola. BCP 84, RFC 3704, "Ingress Filtering for Multihomed Networks." http://tools.ietf.org/html/rfc3704. March 2004.

Bellovin, Steven M., Bill Cheswick, and Angelos D. Keromytis. "Worm Propagation Strategies in an IPv6 Internet." http://www.cs.columbia.edu/~smb/papers/v6worms.pdf. February 2006.

Blunk, L., J. Damas, F. Parent, and A. Robachevsky. RFC 4012, "Routing Policy Specification Language next generation (RPSLng)." http://tools.ietf.org/html/rfc4012. March 2005.

Cisco. "BGP Support for TTL Security Check." http://www.cisco.com/en/US/products/sw/iosswrel/ps5207/products_feature_guide09186a008020b982.html.

De Clercq, J., D. Ooms, S. Prevost, and F. Le Faucheur. RFC 4798, "Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE)." http://tools.ietf.org/html/rfc4798. February 2007.

De Clercq, J., D. Ooms, M. Carugi, and F. Le Faucheur. RFC 4659, "BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN." http://tools.ietf.org/html/rfc4659. September 2006.

Döring, Greg. "IPv6 BGP filter recommendations." http://www.space.net/~gert/RIPE/ipv6-filters.html. August 17, 2008.

Droms, R. (Ed.), J. Bound, B. Volz, T. Lemon, C. Perkins, and M. Carney. RFC 3315, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)." http://tools.ietf.org/html/rfc3315. July 2003.

Gill, V., J. Heasley, D. Meyer, P. Savola, Ed., C. Pignataro. RFC 5082, "The Generalized TTL Security Mechanism (GTSM)." http://tools.ietf.org/html/rfc5082. October 2007 (obsoletes 3682).

Ford, Matthew, Jonathan Stevens, and John Ronan. "Initial Results from an IPv6 Darknet," International Conference on Internet Surveillance and Protection (ICISP'06). August 2006.

Huston, G. RFC 4147, "Proposed Changes to the Format of the IANA IPv6 Registry." http:/ /tools.ietf.org/html/rfc4147. August 2005.

Huston, G. RFC 4177, "Architectural Approaches to Multi-homing for IPv6." http://tools.ietf.org/html/rfc4177. September 2005.

IANA. "IPv6 Global Unicast Address Assignments." http://www.iana.org/assignments/ipv6-unicast-address-assignments.

IANA. "IANA IPv6 Special Purpose Address Registry." http://www.iana.org/assignments/iana-ipv6-special-registry.

Ishihara, K., M. Mukai, R. Hiromi, and M. Mawatari. "Packet Filter and Route Filter Recommendation for IPv6 at xSP routers." http://www.cymru.com/Bogons/ipv6.txt. 2007/06/26.

LACNIC. ftp://ftp.lacnic.net/pub/stats/lacnic/delegated-lacnic-latest.

Lind, M., V. Ksinant, S. Park, A. Baudot, and P. Savola. RFC 4029, "Scenarios and Analysis for Introducing IPv6 into ISP Networks." http://tools.ietf.org/html/rfc4029. March 2005.

McArtor, Gary. "Secure Cisco IOS BGP Template Version 5.5," June 25, 2008. http://www.cymru.com/Documents/secure-bgp-template.html.

Nordmark, E. and T. Li. RFC 4218, "Threats Relating to IPv6 Multihoming Solutions." http://tools.ietf.org/html/rfc4218. October 2005.

RIPE/NCC. ftp://ftp.ripe.net/pub/stats/ripencc/delegated-ripencc-latest.

Yang, Xinyu, Ting Ma, and Yi Shi. "Typical DoS/DDoS Threats under IPv6," International Multi-Conference on Computing in the Global Information Technology (ICCGI'07). March 2007, pp. 55.

# INDEX

## Numerics

**6BONE project, 6**
**6in4 tunnels**
- amplification prevented by, 422
- antispoofing defense, 447
- configuration example, 421
- Linux configuration, 423
- packet structure, 420
- securing, 447-448
- security not inherent in, 421
- viewing status of, 422

**6LoWPAN (IPv6 over Low power Wireless Personal Area Networks), 412**
**6PE tunnels, 109**
**6to4 tunnels**
- 6in4, compared to, 423
- 6TO4 general prefix, 424
- address computation, 424
- addresses, securing, 450
- advantages of, 424
- announcements, 427
- anycast addresses with, 427, 452
- asymmetric routing issue, 427
- configuring in IOS, 424
- detecting, 288-292
- encapsulation details, 425
- hosts acting as routers, 428
- ICMP redirect, securing, 450
- ICS-based latent threat, 461
- ingress ACL for routers, 450
- IPsec with, 453
- ipv6 route command, 425
- Linux configuration of, 427
- loopback interfaces, 427
- MTU settings, 427
- Neighbor Discovery, securing, 450
- network example, 424
- next hop issue, 427
- policy recommendation, 506
- relay ingress ACL, 451-452
- relays, 425, 427
- rogue RAs from, 185
- routers for, 424
- scanning vulnerability, 453
- securing, 450-453
- Unicast RPF with, 451

**6VPE tunnels, 434-436, 459**
**6VPE VRF, 109-110**

## A

**AAA servers, 229-230**
**ACEs (access control entries), 17**
**ACLs (access control lists)**
- applying to interfaces, 147
- BGP with, 93
- blocking inbound router control packets, 263-264
- Cisco-named. *See* IOS ACLs
- defined, 17
- disabling all IPv6 packets, 443
- extended ACLs, 139
- extension header example, 28-29
- filtering compared to, 164
- fragmentation blocking, 47-49
- iACLs, 97, 263-264
- implicit rules, 142-143
- Internet example, IOS, 143-147
- IOS. *See* IOS ACLs
- IPsec external interface example, 337
- IPsec IPv6 remote sites, 342-343
- IPsec IPv6 site-to-site, 340-342
- ISATAP tunnel ingress, 453
- logging issues, 136
- MIPv6, for. *See* MIPv6 filtering
- multicast filtering example, 23
- object group policy configuration, 168-172
- PIX/ASA/FWSM firewall ACLs, 164-168
- receive ACLs, 265
- reflexive, 147-149
- router alert packet blocking, 35
- show command, 167
- show conn command, 168
- show ipv6 access-list command, 469
- stateless nature of, 147
- syntax for, 138-139

**ACS (Access Control Server) servers, 229-230**
**Active Virtual Forwarders (AVFs), 260**
**Active Virtual Gateways (AVGs), 260**

**address spaces.** ***See also*** **addresses**
allocation lists, 87
current list, source for, 132
prefixes, list for blocking, 88
provider independence, 119
reserved space, 131
size compared to IPv4, 4
**addresses**
allowing only allocated, 129
assignment, 113-119
BSD policies, 297
DHCPv6 for assigning, 114-119
documentation, reserved for, 131
EUI-64, 205
forensic tracking of, 483
legitimate IPv6, lists of, 129
loopback, 233-235
Microsoft Windows policies, 296-297
Null 0, 234
prefix policies, 295-296
privacy extension, 205-208
selecting for hosts, issues with, 295-297
SLAAC for assigning, 113
Solaris policies, 297
to deny, list of, 129, 131-132
types available for nodes, 295
unallocated. *See* unallocated addresses
unique-local unicast block, 132
**adoption of IPv6, status of, 6-7**
**advantages of IPv6 over IPv4, list of, 4**
**AHs (Authentication Headers)**
defining RFC, 321
filtering issues, 134
ICV fields, 321
IKE with, 321
IPv6 tunneling example, 346
NAT issues, 325
packet inspection issues, 321
purpose of, 320
SAs with, 324
SPIs, 324
transport mode, 322
tunnel mode, 322
**ALGs (Application Level Gateways), 437, 460**
**alive6 scanning tool, 61**
**amplification attacks, 74-76**
**antispoofing filtering requirement, 128**
**anycast**
6to4 relay allocated addresses, 427
6to4 tunnels using, 452
defined, 4
traffic pattern changes from, 10
**AnyConnect**
address pool usage, 371
advantages of, 369
ASA configuration commands, 370-371
client connection status, 371
configuration steps, 369
dual-stack access, 373
remote connections, 369
split tunneling, 370
**Apple Mac OS X known vulnerabilities, 441**
**application layer attacks, 55**
**Application Level Gateways (ALGs), 437, 460**
**architecture, overview of security changes for IPv6, 9-12**
**ARP (Address Resolution Protocol)**
control plane, role in, 239
NDP used instead of. *See* NDP
not used by IPv6, 187
**ARs (Access Routers), 378**
**AS numbers, private, 103**
**AS path hops, 102**
**ASDM graphical interface, 162**
**atomic IPv6 signatures, 486**
**attackers.** ***See*** **hackers**
**attacks**
amplification, 74-76
DAD based, 191-192
differences between IPv6 and IPv4, 501
distributed DoS. *See* DDoS attacks
denial-of-service. *See* DoS (denial of service) attacks
forged router advertisements, 185, 199-201
fragmentation, 45-49
ICMPv6 based, 20-22
internal, 8
neighbor advertisement, 189
packet-flooding, 74-76
redirection, 194
similarities, IPv4 and IPv6, 500
Smurf attacks, 74-75
vectors for, 7

**authentication**
AAA servers for, 229-230
DHCPv6, of, 209
enabling in IS-IS, 244
headers. *See* AHs
Host-to-Host IPsec for, 326
HSRP, of, 258
HTTP router management interfaces, of, 231
IPsec with OSPFv3, 247
MD5, 247
mobile device issues, 383, 408
NAT issues for, 324-325
OSPFv3-based, 247
proxies, 158
router protocols, of, 241
**Authentication Headers. *See* AHs**
**authorization, 229-230**
**automated reconnaissance, 56-58**
**AUX (auxiliary) interface password, 225**
**availability, 319**
**AVFs (Active Virtual Forwarders), 260**
**AVGs (Active Virtual Gateways), 260**

## B

**best practices, 11**
**BGP (Border Gateway Protocol)**
ACLs with, 93
areas to secure, list of, 91
AS numbers, private, 103
AS path hop length, 102
BTSH, 94
defined, 90
disabling fast external fallover, 104
disabling route-flap dampening, 104
explicitly configured peers, 92
External versus Internal, 91
extreme security measures, 106-107
filtering peering interfaces, 97
global addresses, advantages of, 102
graceful restart configuration, 104
GTSM, 94
IGP with, 106
IPsec with, 93
limiting number of prefixes received, 103
link-local peering, 97-102
logging neighbor activity, 106
loopback addresses, 93
NDP with, 106
next-hop addresses, 99-100
peer availability maximization, 103-106
resets, connection, 105-106
route refresh capability, 105
router configuration example, 94
security approaches, list of, 91
session shared secrets, 92
static content-addressable memory entries, 106
TTL control, 94-95
VPNs, 436
**Binding Updates, 495**
**BitTorrent Teredo vulnerability, 456**
**black holes, 84, 234**
**bogon filtering, 87-90**
**Border Gateway Protocol. *See* BGP**
**botnets**
defined, 80
filtering attacks, 81-82
ingress/egress filtering, 81-82
IPv6 versus IPv4, 81
SYN-Flood attacks, 83
tracing back to attackers, 82-83
**broadcasts, multicast used instead, 74**
**BSD (Berkeley Software Design) OSs**
address selection issues, 297
firewalls for, 303-312
forwarding, stopping, 294
IPFilter, 310-312
ipfirewall, 306-310
known vulnerabilities, table of, 441
neighbor caches, 287
Packet Filter, 304-306
ports, monitoring, 285
tunnel detection, 291-292

## C

**C (Customer) routers, 435**
**Cain & Abel, 182**
**Campus Manager, 472, 494**
**Care of Addresses. *See* CoAs**
**Care of Test Init (CoTI) messages, 383**
**Catalyst Integrated Security Features, 204**

**CBAC (Context-Based Access Control), 149-153**
**CEF (Cisco Express Forwarding) tables, 66**
**Certification Path Advertisements (CPAs), 198**
**Certification Path Solicitations (CPSs), 198**
**CGAs (Cryptographically Generated Addresses), 196-198**
**change management, 493-495**
**Cisco Express Forwarding (CEF) tables, 66**
**Cisco IOS. *See* IOS, Cisco**
**Cisco IOS firewalls. *See* IOS firewalls**
**Cisco IPS 6.1, 486-489**
**Cisco IPsec clients, 364**
**Cisco IPsec VPN client vulnerability, 441**
**Cisco NetFlow Collector, 478**
**Cisco Security Agent. *See* CSA**
**Cisco Security Center (CSC), 221**
**Cisco Security Monitoring, Analysis, and Response System. *See* CS-MARS**
**Cisco Self Defending Network (SDN), 10-11**
**Cisco-named ACLs. *See* IOS ACLs**
**CiscoSecure Access Control Server (ACS) servers, 229-230**
**CiscoView, 472, 494**
**CiscoWorks LMS, 472, 494**
**classes, QoS, 273**
**CLI commands, 270-271**
**CM (CiscoWorks Campus Manager), 472, 494**
**CMDBs (Configuration Management Databases), 493**
**CNs (Correspondent Nodes)**
  authentication issues, 383
  defined, 378
  direct communications mode, 382-385
  indirect mode messages, 381
  MIPv6 filtering of, 395-398
  MN communication links, 379
  spoofing MN bindings, 389
**CoAs (Care of Addresses)**
  authentication issues, 383
  communications model, 380
  defined, 378
  Foreign Agent role of, 379
  HA tracking of, 378
**collaborative security, 11**
**Common Vulnerabilities and Exposures (CVE) database, 442**
**Computer Security Institute (CSI), 8**
**confidentiality, 319. *See also* encryption**
**configuration management, 493-495**
**Configuration Management Databases (CMDBs), 493**
**configured tunnels, 420-423**
**connection interception threats, MIPv6, 388-389**
**console ports, 225**
**consolidated list of recommendations, 508-511**
**consumer grade routers, 111-112**
**Context-Based Access Control (CBAC), 149-153**
**control plane**
  CoPP, 93, 142, 265-269
  defined, 220
  FHRPs, 255-262
  HTTP policing, 268
  policing, 265-269
  role in routers, 262
  show policy-map control-plane command, 267
  SSH policing, 268
  Telnet policing, 268
  vulnerabilities, 239-241
**Control Plane Policing (CoPP), 93, 142, 265-269**
**controlling resources. *See* router resource control**
**CoPP (Control Plane Policing), 93, 142, 265-269**
**Correspondent Nodes. *See* CNs**
**CoTI (Care of Test Init) messages, 383**
**counters, IOS firewall, 155**
**covert channels**
  defined, 17
  extension header padding, 30-32
**CPAs (Certification Path Advertisements), 198**
**CPE (customer premises equipment), 110-113**
**CPSs (Certification Path Solicitations), 198**
**crypto ipsec transform sets, 340**
**Cryptographically Generated Addresses (CGAs), 196-198**
**cryptography. *See* encryption**
**CS (CiscoWorks Common Services), 472**
**CSA (Cisco Security Agent)**
  capabilities of, 313
  dual stack protection with, 443
  IP address tracking, 315
  Network Address Sets, 314-315
  purpose of, 313
  Teredo traffic control, 315
**CSC (Cisco Security Center), 221**
**CSI (Computer Security Institute), 8**

**CS-MARS (Cisco Security Monitoring, Analysis, and Response System)**
- IPv6 events available for rules, 492-493
- NetFlow inputs, 473
- purpose of, 489
- report creation, 491
- rule creation, 491
- user interface of, 490

**Customer (C) routers, 435**
**customer premises equipment, 110-113**
**CVE (Common Vulnerabilities and Exposures) database, 442**

## D

**DAD (Duplicate Address Detection)**
- attacks on, 192
- messages with unspecified addresses, 131
- normal behavior, 190
- privacy addresses with, 206
- protection mechanisms, 195
- purpose of, 174-175, 190

**darknets, 84**
**Data Link Layer. *See* Layer 2**
**data traffic plane, 220**
**DDNS (Dynamic DNS), 484**
**DDoS (distributed denial of service) attacks**
- black hole defense, 84
- botnets for, 80-83
- darknet defense, 84
- defined, 80
- ingress/egress filtering, 81-82
- IPv6 versus IPv4, 81
- SYN-Flood attacks, 83
- tracing back to attackers, 82-83

**debug command**
- ACL setup, 149
- options for firewalls, 157

**default gateways**
- FHRP redundancy mechanism vulnerabilities, 255
- GLBPv6 for, 260-262
- HSRP for, 257-259
- NUD redundancy mechanism vulnerabilities, 255-257
- router roles for, 260

**Default Router Preference (DRP), 256**
**defense in depth strategy, ICMPv6 filtering, 283**
**definition of IPv6, 3**
**denial of service attacks. *See* DoS attacks**
**deployment timeline, global, 6**
**destination address field, IPv6 headers, 16**
**Destination Options headers (DOHs)**
- ACL options syntax, 35
- attacks in padding, 30-32
- guidelines for, 25
- Mobile IPv6, 379-381, 393
- structure of, 29

**devices**
- Device Fault Manager, 472
- management of. *See* management systems
- network. *See* network devices
- performance monitoring overview, 469

**DFM (CiscoWorks Device Fault Manager), 472**
**DHCP Unique Identifiers (DUIDs), 117-118, 208**
**DHCPv6**
- address assignment by, 114-119
- ADVERTISE messages, 209
- attack targets, 22
- authentication, 209
- differences with v4, 208
- DoS threats to, 210-212
- DUIDs, 117-118, 208
- flags, 210
- forensics with, 484
- IOS routers as relays, 210
- misinformation threats, 210, 213
- mitigating attacks, 211-213
- ports used by, 209
- purpose of, 208
- RENEW messages, 210
- REPLY messages, 209
- REQUEST messages, 209
- rogue server threats, 210, 213
- scanning threats, 210, 213
- snooping tool, 204
- SOLICIT messages, 209
- starvation threats, 210-211
- stateful message exchanges, 209-210
- UIDs, 117-119, 208

**differences between IPv4 and IPv6, 501**
**disabling IPv6, 443**
**disabling unnecessary network services, 222-224**

**distributed denial of service attacks.** ***See*** **DDoS attacks**

**DMVPN (Dynamic Multi-Point Virtual Private Network)**

- central router configuration, 351-352
- dual-stack functionality, 353
- IPsec connection status commands, 355-359
- NHRP with, 350
- purpose of, 349
- remote-site router configuration, 352-353
- topology example for, 350
- verifying at the hub, 353-354
- verifying at the spoke, 359-361

**DNS (Domain Name System)**

- dual stack inquiries to, 418
- dynamic, 484
- importance for IPv6, 502
- ISATAP attacks through, 454
- Microsoft Windows XP default addresses, 300
- privacy address issues, 484
- scans of, 56
- Security (DNSSEC), 502

**DOHs.** ***See*** **Destination Options headers**

**DoS (denial of service) attacks**

- DAD based, 191-192
- DHCPv6 based, 210-212
- distributed. *See* DDoS attacks
- forged RAs, 185
- HSRP vulnerability to, 257
- ICMPv6 based attacks, 21
- IOS firewalls for, 150-152
- ip directed broadcasts, disabling, 74
- MIPv6, on, 390
- Morris worm, 74
- multicasting for, 22
- packet-flooding attacks, 74-77

**dos-new-ipv6, 192**

**DRP (Default Router Preference), 256**

**DSMIPv6 (Dual Stack MIPv6), 408**

**dual stacks**

- address selection issues, 295
- application attacks, 55
- application layer with, 418
- Cisco Security Agent for protection, 443
- deciding between IPv6 and IPv4, 418
- defined as transition technique, 417
- disabling IPv6, 443
- dual routing tables required, 418
- exploiting, 440-443
- host operating system issues, 281
- IPv6 preferred over IPv4, 418
- ISATAP for hosts. *See* ISATAP
- known vulnerabilities, table of, 441
- latent threats, 441
- Layer 2 types, 417
- MAC OS X latent threat mechanism, 440
- mechanism depiction, 417
- memory consumption issue, 418
- no split tunneling vulnerability, 441
- personal firewalls for protection, 443
- preferred transition technique, 417
- protection methods for, 443-444
- recommendations, list of, 510
- strategy, 4-5
- transport protocols with, 418
- vulnerabilities, 440-443

**DUIDs (DHCP Unique Identifiers), 117-118, 208**

**Duplicate Address Detection.** ***See*** **DAD**

**dynamic crypto maps, 338-339**

**Dynamic DNS (DDNS), 484**

**dynamic multipoint VPNs.** ***See*** **DMVPN**

**dynamic tunnels**

- 6to4. *See* 6to4 tunnels
- defined, 420
- ISATAP. *See* ISATAP
- securing, 449
- Teredo. *See* Teredo

## E

**early adopter strategy, 7**

**EBGP.** ***See*** **BGP**

**edge routers (ERs), 110**

**EIGRPv6, 242-244**

**enable secret command, 225**

**Encapsulated Security Payloads.** ***See*** **ESPs**

**encryption**

- HSRP authentication, 258
- keygen tokens, 384
- neighbor discovery, of. *See* SEND
- overview of technologies for, 319
- passwords, of, 225

**endpoint security, 215.** ***See also*** **host security**

**ERs (edge routers), 110**
**ESPs (Encapsulated Security Payloads)**
capabilities of, 320
defining RFC, 320
IKE with, 321
IPv6 over IPv4 example, 329
IPv6 tunneling example, 346
NAT issues, 325
null encryption, 321
purpose of, 320
SAs for, 324
SPIs, 324
tunnel mode, 322
**EUI-64 (Extended Unique Identifier 64)**
address configuration, 159, 205
ISATAP generation of, 429
**explicitly configured BGP peers, 92**
**extended ACLs, 139**
**extension headers**
ACL example for, 28-29
application layer attacks using, 55
chain size vulnerability, 28
defining RFC, 24
Destination Options headers, 25, 29-32
dual-stack attacks, 55
formats, 24
fragment headers, 25, 43, 47-52
future of, 507
fuzzing, 33
Hop-by-Hop Options header, 25, 29-32
inspection issues, 27
IPsec, 320-321
mobile, types of, 379
Next Header field. *See* NH
next-header numbers, 25, 27
NH check problem, 47
order of, 25
overview, 24-25
policies for, 504
purpose of, 24
router alert attacks, 33-36
routing. *See* routing headers
rules of, 25
types requiring special attention, 28
unknown headers, 52-54
vulnerabilities, 28

# F

**fake_advertise6, 189**
**fake_router6, 185, 200, 202**
**FAs (Foreign Agents), 379**
**fast external fallover, 104**
**Fast Handover for Mobile IPv6 (FMIPv6), 407**
**FCAPS model, 467**
**FHRPs (first-hop redundancy protocols)**
GLBPv6, 260-262
HSRPv6, 257-259
NHD (Neighbor Unreachability Detection), 255-257
purpose of, 255
**FIB (Forwarding Information Base), 119**
**filtering. *See also* firewalls**
ACLs compared to, 164
allocated addresses, permitting, 129
BGP peer interfaces, 97
deny lists for addresses, 129-132
extended ACLs for, 139
header issues, 133-134
ICMPv6 messages, 20-22
ingress/egress. *See* ingress/egress filtering
internally allocated addresses at perimeter, 132
link-local addresses, 131
MIPv6. *See* MIPv6 filtering
multicasting address guidelines, 131
PIX/ASA/FWSM firewalls for, 164-166
reserved space, 131
unallocated addresses at firewalls, 128-133
**firewalls. *See also* filtering**
allocated addresses, permitting, 129
antispoofing requirement, 128
ASA. *See* PIX/ASA/FWSM firewalls
availability of, 12
basic policy rule, 128
BSD, 303-312
Cisco IOS. *See* IOS firewalls
deny lists for addresses, 129-132
FWSM. *See* PIX/ASA/FWSM firewalls
header issues, 133-134
host. *See* host firewalls
ICMP filtering issues, 18
ICMP unreachable messages, 136
inspecting tunneled traffic, 134-135
internally allocated addresses, blocking, 132

IPFilter, 310-313
ipfirewall, 306-310
IPv6 overview, 128
IPv6 versus IPv4, implications, 502
IPv6-only, advantages of adding, 128
Layer 2, 135
Linux, 301-303
logging, 136
Microsoft Windows Firewall, 298, 300-301
multicasting address guidelines, 131
NAT with, 136-138
NetFilter, 301-303
Packet Filter, 304-306
performance issues, 136
PIX. *See* PIX/ASA/FWSM firewalls
policy recommendations for, 504
show ipv6 interface command, 177
Solaris, 312-313
unallocated addresses, filtering, 128-133

**first-hop redundancy protocols. *See* FHRPs**

**Flexible Packet Matching (FPM), 457**

**flow label field, IPv6 headers, 16**

**flows. *See* NetFlow tool**

**FMIPv6 (Fast Handover for Mobile IPv6), 407**

**Foreign Agents (FAs), 379**

**forensics, 481-483**

**forwarding**
BSD, 294
Forwarding Information Base (FIB), 119
Linux, 293-294
Microsoft Windows, 293
Solaris, 294
vulnerabilities from, 292

**FPM (Flexible Packet Matching), 457**

**fragment headers**
ACLs for blocking, 47-49
attacks with, 45-49
end system attacks, 49
guidelines for, 25
NH check problem, 47
virtual fragment reassembly, 49-52

**fragmentation. *See also* fragment headers**
attack tools, 44
attacks with, 45-49
fragment IDs, 43
ICMPv6 Packet Too Big messages, 45
overview of issues, 43-45
PMTUD sizing, 45
protection, 172
very small fragments suspect, 45
virtual fragment reassembly, 49-52

**FreeBSD. *See* BSD**

**FreeRADIUS, 230**

**FTP, Secure. *See* SFTP**

**future of IPv6, 506-508**

**fuzzing headers, 33**

**FWBuilder, 301**

## G

**Gateway Load Balancing Protocol (GLBPv6), 260-262**

**Generalized TTL-based Security Mechanism (GTSM), 94**

**GLBPv6 (Gateway Load Balancing Protocol), 260-262**

**global addresses, defined, 430**

**Graceful Restart, BGP, 104**

**GRE (generic routing encapsulation) tunnels, 420-421, 447**

**GRE tunnels, 333, 350**

**GTSM (Generalized TTL-based Security Mechanism), 94**

## H

**hackers, 8-9. *See* also attacks**

**hacking tunnels, 444-446**

**handover time, 378**

**hard resets, 105**

**hardening network devices**
disabling unnecessary services, 222-224
FHRP security. *See* FHRPs
interface hardening, 223-224
limiting router access. *See* router access management
management system vulnerabilities. *See* management systems
overview of, 219
quality issues. *See* QoS
recommendations, list of, 509
resources. *See* router resource control
routing protocol threats. *See* routing protocols

**HAs (Home Agents)**
defined, 378
filtering at, 402-406
IPsec with, 390-392
registration by MNs, 379
turning off MIPv6, 392
**HBA (Hash Based Addresses), 121**
**HDLC (High-Level Data Link Control), 213**
**headers**
Authentication Header, 134
covert channels, 17
defining RFC, 16
destination address field, 16
extension. *See* extension headers
filtering issues, 133-134
flow label field, 16
fragment. *See* fragment headers
fuzzing, 33
hop limit field, 16
IPsec, 320-321
IPv4 versus IPv6, 133
IPv6 similarities to IPv4, 500
Next Header field. *See* NH
option extension headers, 133
payload length field, 16
routing. *See* routing headers
source address field, 16
traffic class field, 16
unknown headers, 52-54
version field, 16
vulnerabilities, relation to, 17
**Hierarchical Mobile IPv6 (HMIPv6), 407**
**High-Level Data Link Control (HDLC), 213**
**HMIPv6 (Hierarchical Mobile IPv6), 407**
**HoAs (Home Addresses)**
defined, 378
determination mechanism, 381
direct communications mode, 382-385
goal for, 378
HA relationship to, 378
**Home Agents. *See* HAs**
**Home Test Init (HoTI) messages, 383, 386**
**hop limits**
header field, 16
hop-limit command, 224
ICMPv6 issues, 19-20
maximum value, 195
**Hop-by-Hop Options headers**
attacks in padding, 30-32
defined options of, 30
guidelines for, 25
number of options allowed, 30
router alert attacks, 33-36
structure of, 29
testing example, 30-32
**host firewalls**
BSD, 303-312
Linux, 301-303
Microsoft Windows Firewall, 298-301
recommended for IPv6, 297
Solaris, 312-313
**host security**
address selection issues, 295-297
application ports, 284
BSD ports, 285
Cisco Security Agent. *See* CSA
dual-stack issues, 281
firewalls. *See* host firewalls
forensic challenges, 483
forwarding by hosts, stopping, 292-295
ICMPv6 processing, 282-283
ISATAP tunnel detection, 288-289
Linux ports, 284
Microsoft Windows ports, 284
neighbor caches, 285-287
overview of, 281-282
policies for, 505
recommendations, list of, 509
servers with multiple network interfaces, 292
Solaris ports, 285
Teredo tunnel detection, 289-290
tunnel detection, 287-292
**Host-to-Host IPsec, 326-328**
**hosts. *See also* host security**
defined, 4
firewalls of. *See* host firewalls
**Hot Standby Routing Protocol. *See* HSRP**
**HoTI (Home Test Init) messages, 383, 386**
**HP OpenView NNM (Network Node Manager), 472**
**HSRP (Hot Standby Routing Protocol), 257-259**
**HTTP (Hypertext Transport Protocol)**
access to routers, 230-233
policing control traffic, 268
router server status, showing, 232

## I

**iACLs (infrastructure ACLs)**
- blocking inbound router control packets, 263-264
- purpose of, 97

**IBGP. *See* BGP**

**ICMPv6 (Internet Control Message Protocol version 6)**
- ACL syntax for, 165
- ACLs, specifying for, 139
- attacks and mitigation, 20-22
- blocking all packets, 18
- Cisco planned feature set for, 204
- defense in depth strategy, 283
- defining RFC, 17
- DoS attacks based on, 21
- Duplicate Address Detection, 190, 192
- Echo Request packet attacks, 20-21
- error messages, 19, 21-22, 505
- error-interval command, 22
- filtering by function, 21
- firewall policy example, 166
- forged RAs, 185-186
- functions provided by, 18-19
- guidelines for host packet monitoring, 283
- history of, 18
- hop limit issues, 19-20, 195
- host processing, 282-283
- icmp error-interval command, 269
- IETF working group on, 203
- interface policy, 165
- Layer 2 vulnerability overview, 182
- message structure, 19
- MIPv6 filtering of, 393-394
- NDP attack mitigation, 201-204
- neighbor discovery, 187-190. *See also* neighbors
- Neighbor functions, 18
- normal router advertisement mechanism, 183
- Packet Too Big messages, 45
- packet-flooding attack vulnerability, 76
- permitted message type recommendations, 282
- protocol protection mechanisms, 195-199
- QoS traffic limiting, 271
- redirection issues, 193-195
- reducing attack scope, 203
- rogue nonmalicious RAs, 185
- rogue RA attack detection, 199-200
- Router Advertisements, 183. *See also* RAs
- SEND, 196-199
- SLACC issues, 183, 185-186
- source address restrictions, 195
- source and destination address issues, 20
- stateless autoconfiguration addresses, 20
- type numbers, 19
- unallocated message types, 20
- unreachable messages, 136
- unreachable packets, preventing, 223

**ICS (Internet Connection Sharing), 461**

**IDS (inline intrusion detection), 158**

**IDSs (Intrusion Detection Systems), 485**

**IEE 802.16e, 411**

**IETF (Internet Engineering Task Force), 3**

**ifconfig command for tunnel detection, 291-292**

**IGPs (interior gateway protocols), 106. *See also* IS-IS**

**IKE (Internet Key Exchange)**
- configuration for IPv6 tunnels, 344-345
- ICV fields, 321
- IPv6 over IPv4 configuration, 329
- mechanics of, 323
- purpose of, 321-322
- SAs with, 324
- SPDs with, 323
- tunneling parameters, viewing, 333
- versions, 324

**image verification of IOS, 221-222**

**indirect mode mobile tunneling, 381**

**Information Technology Infrastructure Library (ITIL), 493**

**infrastructure ACLs. *See* iACLs**

**ingress/egress filtering**
- address prefix list for, 88-89
- allocated addresses, on, 85-86
- bogon filtering, 87-90
- DDoS attacks, 81-82
- importance of, 85
- Internet Routing Registries, 90
- locations for, 85
- spoofing attacks, preventing, 66
- unallocated addresses, 87

**injection attacks on tunnels, 444-446**

**inspect command, IOS firewalls, 150-153**

**inspect statistics command, IOS firewalls, 156**
**inspecting traffic. *See* filtering; firewalls**
**integrity of communications, 319. *See also* MD5**
**interfaces**
AUX, passwords for, 225
configured tunnel properties, 423
disabling routing services on, 235
hardening, 223-224
loopback, 233-235
management network, 234-235
monitoring, 468-469
Null 0, 234
OSPFv3, 250
passive-interface command, 240
remote, viewing, 365-367
show ipv6 interface command, 177
**interior gateway protocols (IGPs), 106**
**interior routing protocol threats. *See* routing protocols**
**Intermediate System-to-Intermediate System. *See* IS-IS**
**Internet Connection Sharing (ICS), 461**
**Internet Control Message Protocol. *See* ICMPv6**
**Internet Engineering Task Force (IETF), 3**
**Internet Key Exchange. *See* IKE**
**Internet Protocol (IP) hourglass, 499**
**Internet Routing Protocol Attack Suite (IRPAS), 257**
**Internet Routing Registries (IRRs), 90**
**Intra-Site Automatic Tunnel Addressing Protocol. *See* ISATAP**
**IOS, Cisco**
6in4 tunnel configuration, 421
ACLs. *See* IOS ACLs
firewalls. *See* IOS firewalls
image verification, 221-222
ISATAP configuration, 430
Safe Harbor testing, 221
SEND CGA configuration, 198
version selection, 220-222
**IOS ACLs**
clear command, 139
CoPP/CPPr conflicts, 142
debug command, 149
explicit denies, 143
extended ACLs, 139
extension header example, 28-29
firewall configuration example, 154
firewalls, configuring for, 150-153
ICMP with, 139
implicit neighbor discovery, 142
implicit rules, 142-143
Internet ACL example, 143-147
logging, 140
named access list style, 138
NDP for MAC mapping, 142
PMTUD requirement, 143
processor burden from, 139
purpose of, 138
reflexive, 147-149
show command, 139
standard IPv6 ACLs, 138
stateless nature of, 147
switches with, 139
syntax for, 138-139
TCP with, 139
traffic-filter commands, 147
UDP with, 139
undetermined-transport keyword, 52-54
uploading with SCP, 140, 142
**IOS firewalls**
audit trail inspection, 153-155
authentication proxy with, 158
CBAC, 149-153
configuration example, 153-157
configuring, 150-153
counters, clearing, 156
counters, viewing, 155
CPU utilization, 150
debug options, 157
defining inspection policies, 150
DoS attack monitoring, 150-152
hash table size extension, 158
high water mark settings, 152
IDS with, 158
inspect statistics command, 156
inspectable types of packets, 151
IPv6 support, first version with, 149
Port-to-Application, 157-158
routing header inspection, 151
sessions, removing, 156
show command, 154
stateful features, 149
TCP connection settings, 151

- UDP connection settings, 152
- VFR feature, 150

**IP (Internet Protocol) hourglass, 499**
**IP Next Generation (IPng), 3**
**IP phones, 270**
**IP Stack Integrity Checker (ISIC), 33**
**ip6tables firewall, 301-302**
**ip6wall, 301**
**IPFilter (ipf), 310-313**
**ipfirewall (ipfw), 306-310**
**ip-firewalling, 301**
**IPM (CiscoWorks Internetwork Performance Monitor), 472**
**IPng (IP Next Generation), 3**
**IPsec (IP Security)**

- 6to4 tunnels with, 453
- ACLs for IPv6 site-to-site, 340-342
- active interfaces, viewing, 347-348
- Authentication Headers. *See* AHs
- BGP with, 93
- Cisco IOS router configuration, 328
- Cisco IPsec clients, 364
- defined, 320
- dynamic crypto maps, 338-339
- dynamic multipoint VPN. *See* DMVPN
- Encapsulated Security Payloads. *See* ESPs
- encryption options, 349
- extension headers for, 320-321
- GRE tunnels, 333
- history of, 320
- Host-to-Host, 326-328
- IKE configuration, IPv6 tunneling, 344-345
- implementation issues in IPv6, 325-326
- inspecting traffic, 134
- Internet Key Exchange. *See* IKE
- IPS, obscurity to, 326
- IPv6 over IPv4 configuration example, 329-337
- IPv6 site-to-site configuration, 339-349
- IPv6 tunnel status, 343-344
- ISAKMP policy, 340
- IS-IS issues, 333
- Linux support for, 327
- man-in-the-middle attack prevention with, 321
- Microsoft SDI, 327
- MIPv6 use of, 390-392
- modes of operation, 322
- NAT issues, 324-325
- OSPFv3 authentication with, 247-251
- packet inspection issues, 321
- perimeter security changes from IPv6, 502
- policy recommendations, 506
- recommendations, list of, 510
- remote access configuration, 361-368
- remote site configuration, 342-343
- routing tables, viewing, 332, 348-349
- SADBs, 324
- SAs with, 324, 334
- session information, viewing, 347
- site-to-site configuration overview, 328-329
- SPDs with, 323
- SPIs, 247, 324
- state verification, 332
- transport mode, 322
- tunnel mode, 322
- Unicast RPF with, 337
- vendor support for, 327

**IPSs (Intrusion Prevention Systems)**

- atomic IPv6 signatures, 486-489
- Cisco IPS 6.1, 486-487
- importance of, 485
- IPsec, inability to monitor, 326
- NDP packet inspection, 486
- notifications, 487
- policy recommendation for, 504
- Risk Ratings, 486
- viewing events, 488

**IPv4, shortcomings of, 3**
**IPv4 protocol 41, 134**
**IPv4 with IPv6. *See* dual stacks**
**IPv6 over IPv4 IPsec configuration example, 329-337**
**IPv6 over IPv4 tunnels between PE routers, 108**
**IRPAS (Internet Routing Protocol Attack Suite), 257**
**IRRs (Internet Routing Registries), 90**
**ISAKMP (Internet Security Association and Key Management Protocol)**

- IPv6 tunnel configuration, 344-345
- policy, 340
- state information, viewing, 333

**ISATAP (Intra-Site Automatic Tunnel Addressing Protocol)**

- ACLs for, 453
- blocking, 455

BSD, detecting tunnels in, 292
client-side configuration, 429
defined, 428
detecting tunnels in, 288-292
DNS cache poisoning, 454
encapsulation of IPv6 packets, 428
EUI-64, 429
global unicast address generation, 429
initiating connections, 364
IOS router configuration, 430
latent threats of, 454-455, 461
Linux, detecting tunnels in, 291
no ipv6 nd suppress-ra command, 430
pinging remote clients, 367
potential routers lists (PRLs), 429
purpose of, 368
remote user access with, 362
Router Advertisements, 429
Router Solicitations, 429
routers, configuration example, 363
routers, connecting to, 368
routing tables, 367
scanning vulnerability, 455
securing, 453-455
theft of service, 454
tunnel detection, 288-292
tunnel status, showing, 363-364
Unicast RPF checks with, 453
verifying connections, 365
vulnerabilities of, 453
Windows host configuration, 429

**ISIC (IP Stack Integrity Checker), 33**

**IS-IS (Intermediate System-to-Intermediate System)**
defined, 244
IGP security recommendations, 106
MD5 authentication, enabling, 244
neighbor configuration, 245
router types, 244
TLVs, 244
topology modes, 244
tunneling issues, 333

**ISP-level security. *See* server provider security**

**ITIL (Information Technology Infrastructure Library), 493**

# J

**jumbograms, 4**

# K

**KAME project, 6**

**keygen tokens, 384**

# L

**Label Distribution Protocol (LDP), 110**

**label switched paths (LSPs), 435-436**

**LAN (Local Area Network) security policies, 505. *See also* local network security**

**large-scale Internet threats**
DDoS attacks, 80-84
first historic attack, 74
Morris worm, 74
packet flooding, 74-77
worms, 74, 77-80

**latent threats**
6to4, 461
blocking all IPv6 traffic, 462
defined, 417, 441
DNS with ISATAP, 461
host IPv6 stack, disabling, 461
ICS-based, 461
ISATAP, to, 454-455
list of, 460
MAC OS X example, 440
native IPv6 deployment defense, 461
RA based, 461
reducing impact of, 461
Teredo-based, 455, 461
wireless hotspot issue, 461-462

**Layer 2**
attack tools, 182
dsniff tool, 181
Duplicate Address Detection, 190, 192
firewalls, 135
forged RAs, 185-186
ICMPv6 vulnerability overview, 182
importance of, 181-182

neighbor discovery, 187-190
normal router advertisement mechanism, 183
point-to-point links, 213-215
protocol vulnerabilities, 181
redirection issues, 193-195
rogue nonmalicious RAs, 185
Router Advertisements, 183. *See also* RAs
SLACC issues, 183-186
social vulnerabilities, 181
STP vulnerability, 181
traffic diversion tools, 181
types for IPv4 and IPv6, 417

**Layer 3**
signaling protocols for, 239
spoofing, 65-69

**Layer 4 spoofing, 65-69**

**LDP (Label Distribution Protocol), 110**

**link-local addresses**
filtering issues, 131
PIX/ASA/FWSM configuration, 159

**link-local peering, 97-102**

**Linux**
6in4 tunnel configuration, 423
6to4 tunnel configuration, 427
firewalls, 301-303
forwarding, stopping, 293-294
IPsec support, 327
IPv6 enabled by default, 440
known vulnerabilities, table of, 441
neighbor caches, 286
NetFilter firewall, 301-303
netstat command, 284
nmap command, 285
ports used by, 284
rpcinfo command, 284
sysctl command, 293
tunnel detection, 290-291

**LMAs (Localized Mobility Anchors), 408**

**LMS (CiscoWorks LAN Management Solution), 472, 494**

**local addresses, defined, 430. *See also* addresses**

**local network security**
attack tools, 182
Cisco planned feature set for, 204
DHCP issues. *See* DHCPv6
dsniff tool, 181
Duplicate Address Detection, 190, 192
forged RAs, 185-186
ICMPv6 attack mitigation, 201-204
ICMPv6 protocol protection mechanisms, 195-199
ICMPv6 vulnerability overview, 182
Layer 2, importance of, 181-182
neighbor discovery, 187-190
normal router advertisement mechanism, 183
point-to-point links, 213-215
privacy extension addresses, 205-208
protocol vulnerabilities, 181
recommendations, list of, 509
redirection issues, 193-195
rogue nonmalicious RAs, 185
rogue RA attack detection, 199-200
Router Advertisements, 183. *See also* RAs
SEND, 196-199
SLAAC issues, 183, 185-186
social vulnerabilities, 181
traffic diversion tools, 181

**Localized Mobility Anchors (LMAs), 408**

**logging**
IOS ACL based, 140
IOS firewalls inspect audit-trail command, 153, 155
packets blocked at firewalls, 136
syslog, 478-481

**login monitoring of routers, 228-229**

**loopback addresses, 93**

**loopback interfaces, 233-235, 427**

**LSPs (label switched paths), 435-436**

# M

**MAC addresses**
forensics issues, 485
mapping with NDP, 142
privacy extension addresses with, 205
SEND discovery method, 197

**Mac OS X, 440-441**

**malicious users. *See* hackers**

**management**
access, firewall configuration for, 161-162
data traffic plane, 220
network. *See* network management
plane, protecting, 224

security configuration, of, 493-495
tunnels, of, 482-483
**management systems**
disabling routing services on interfaces, 235
loopback interfaces, 233-235
management network interfaces, 234-235
MIBs, 469-470
MPP, 235
Null 0 interfaces, 234
policy recommendations, 506
recommendations, list of, 511
SNMP protocol, 235
vulnerabilities overview, 233
**MANETs (Mobile Ad Hoc Networks), 411-412**
**man-in-the-middle attacks (MITMs)**
IPsec for preventing, 321
Layer 2 vulnerabilities for, 181
MIPv6, attacks on, 387-388
**MAPs (Mobility Anchor Points), 407**
**maximum transmission units (MTUs), 183, 421**
**MD5 (message digest algorithm 5)**
defined, 221
EIGRPv6 configuration, 242
GLBP messages, 260-261
HSRP authentication with, 257-258
IOS image verification with, 221
IS-IS configuration, 244
OSPFv3 authentication with, 247
SPI key creation, 247
**mean time between failure (MTBF), 468**
**mean time to repair (MTTR), 468**
**Meeting Space, 456**
**message digest algorithm 5. *See* MD5**
**MEXT Working Group, 407**
**MIBs (Management Information Bases), 469-470**
**microsegmentation, 203**
**Microsoft Server and Domain Isolation (SDI), 327**
**Microsoft Windows**
6to4 tunnels, detecting, 288
address selection issues, 296-297
disabling IPv6, 443
Firewall, 298-301
forwarding, stopping, 293
ICS rogue RAs, 185-186
ISATAP configuration, 429
ISATAP tunnel detection, 288-289
known vulnerabilities, table of, 441
neighbor caches, 286
netstat command, 284
Portproxy, 301
ports used by, 284
Teredo tunnels. *See* Teredo
tunnel detection, 287-290
Vista, IPv6 enabled by default, 440
XP DNS server addresses, 300
**migration, 9-12**
**MIPSHOP Working Group, 407**
**MIPv6 (Mobile IPv6)**
access routers, 378
address determination for HoAs, 381
architectural changes required by, 10
authentication issues, 383, 408
binding MNs to HAs, 379
blocking, 392
Care of Addresses. *See* CoAs
communications model, 378
compared to IPv4, 4
connection interception threats, 388-389
Correspondent Nodes. *See* CNs
Destination Option Headers, 379, 381, 393
Diameter protocol, 409
direct communications mode, 382-385
DoS attacks on, 390
DSMIPv6, 408
encryption issues, 384
extension headers used by, 379
filtering. *See* MIPv6 filtering
FMIPv6, 407
forwarding system, 378
goal of, 378
handover time, 378
HMIPv6, 407
Home Addresses. *See* HoAs
Home Agents. *See* HAs
indirect communications model, 380-381
IPsec with, 390-392
IPv4 compared to, 379
keygen tokens, 384
MANETs, 411-412
man-in-the-middle attacks on, 387-388
MAPs, 407
media security, 386
messages, overview of, 379, 381

MEXT Working Group, 407
MIPSHOP Working Group, 407
Mobile Nodes. *See* MNs
Mobile WiMax, 411
Mobility Option Headers, 379
NEMO, 409-410
NETLMM Working Group, 408
network layer mobility requirement, 377
operational details of, 378-379
PANA protocol, 409
perimeter security changes from, 503
purpose of, 377
RADIUS protocol, 409
recommendations, list of, 510
requirements for foreign networks, 399
RH2 extension headers, 379-380
rogue home agents, 386
RR procedure, 382-385
software protection, 386
spoofing MN-to-CN bindings, 389
THC spoofing attacks, 388

**MIPv6 filtering**
ACL issues, 393
ACL examples, 394
blocking MIPv6, 392
CN, filtering at, 395-398
Destination Option Headers, 393
firewall issues, 392
foreign network issues, 398-401
goals of, 392
HAs, at, 402-406
ICMPv6, 393-394
MNs, filtering at, 398-401
mobility extension headers, 393
RH0, 393
types of messages to filter, 392

**MITM (man-in-the-middle) attacks**
IPsec for preventing, 321
Layer 2 vulnerabilities for, 181
MIPv6, 387-388

**MLD (Multicast Listener Discovery)**
filtering, 23
link-local address filtering, 20
purpose of, 18

**MNs (Mobile Nodes)**
addresses of, 378
authentication issues, 383
CNs with, 379
communications model, 378
direct communications mode, 382-385
filtering of, 398-401
handover time, 378
HoA address determination, 381
indirect mode messages, 381
IPsec with, 390-392
registration with HAs, 379
requirements for functioning, 399
spoofing CN bindings, 389

**Mobile Ad Hoc Networks (MANETs), 411-412**

**Mobile IPv6. *See* MIPv6**

**Mobile WiMax, 411**

**Mobility Anchor Points (MAPs), 407**

**Mobility Option Headers, 379, 393**

**Modular QoS CLI (MQC) commands, 270-271**

**monitoring**
CiscoWorks LMS, 472
HP OpenView, 472
Multi-Router Traffic Grapher, 471
NetFlow tool, 472-474, 476-478
recommendations, list of, 511
router interfaces, 468-469
security. *See* security monitoring
SNMP MIBs for, 469-470
syslog, 478-481

**Morris worm, 74**

**MPLS (Multiprotocol Label Switching)**
defined, 434
IPv6 over IPv4, 107-110
VPNs, 435-436

**MQC (Modular QoS CLI) commands, 270-271**

**MRD (Multicast Router Discovery), 18**

**MRTG (Multi-Router Traffic Grapher), 471**

**MTBF (mean time between failure), 468**

**MTTR (mean time to repair), 468**

**MTUs (maximum transmission units)**
defined, 183
setting in IOS, 421
PMTUD, 18, 45, 143, 164

**Multicast Listener Discovery. *See* MLD**

**Multicast Router Discovery (MRD), 18**

**multicasting**
ACL example, 23
addresses, table of vulnerable, 59
DoS attacks based on, 22, 74, 76

filtering guidelines, 131
MLD, 18, 20, 23
perimeter security for, 502, 505
reconnaissance with, 59-61
security overview, 22-23
source address inspection, 23

**multihoming**
provider independent address prefixes, 65
security issues, 119-122

**multipoint GRE tunnels, 350**

**Multiprotocol Label Switching.** ***See*** **MPLS**

**Multi-Router Traffic Grapher, 471**

# N

**NAC (Network Admission Control), 484**

**NACL (Network Address Control), 314-315**

**NAPT-PT (Network Address Port Translation-Protocol Translation), 438**

**NAs (Neighbor Advertisements)**
defined, 187
detecting, 201
dynamic inspection of, 204
spoofing, 188-189

**NAT (Network Address Translation)**
firewalls with, 137-138
IPsec issues, 324-325
IPv6 security issues from, 5-6
Protocol Translation, 437, 459-460
purpose of, 136
Teredo tunnels with, 430-431
tunnels, interference with, 421

**NAT-PT (Network Address Translation-Protocol Translation), 437, 459-460**

**NCM (CiscoWorks Network Compliance Manager), 494**

**ndd command (Solaris), 312-313**

**NDP (Neighbor Discovery Protocol)**
atomic signature inspection, 486
attack mitigation, 201-204
control plane, roll in, 239
CoPP/CPPr conflicts, 143
detecting attacks, 201
duplicate address detection, 174-175
improvement over ARP, 188
message types, 187
monitoring, 199-201
NA messages, 187
neighbor commands, 176-177
NS messages, 187
NUD use of, 255
PIX/ASA/FWSM firewalls, 174-177
prefix scanning attacks, 190
protection mechanisms, 195
purpose of, 142
rafixd for attack mitigation, 202-203
reachability confirmation timers, 177
reducing attack target scope, 203
SEND extension to, 196-198
sniffers, 61
spoofing, 188-189
test, 174
turning off, 106
Windows neighbor caches post-attack, 190
Windows neighbor caches pre-attack, 189

**ndp utility (BSD OSs), 287**

**NDPMon utility, 199-201**

**neighbor caches**
attacks based on, 62
BSD, checking in, 287
Linux, checking in, 286
Microsoft Windows, checking in, 286
Solaris, checking in, 287
vulnerability of, 285

**neighbor discovery**
6to4 tunnels, blocking for, 450
IOS ACLs, implicit, 142
protocol. *See* NDP
spoofing of, 187-190

**Neighbor Discovery Protocol.** ***See*** **NDP**

**Neighbor Unreachability Detection (NUD), 255-257**

**neighbors**
caches. *See* neighbor caches
discovery. *See* neighbor discovery; NDP
EIGRPv6 for viewing configuration of, 243
ICMPv6 Neighbor functions, 18
ip neighbor show command, Linux, 286
IS-IS configuration, 245
NA messages, 187
NS messages, 187
OSPFv3 IPsec neighbor state, 249
spoofing, 188-189

Windows neighbor caches post-attack, 190
Windows neighbor caches pre-attack, 189
**NEMO (Network Mobility) ), 409-410, 503**
**NetFilter firewall, 301-303**
**NetFlow tool, 83, 472-478**
**netsh command**
firewall commands, 298
ISATAP router configuration with, 429
privacy address configuration, 207
show neighbors command, 286
tunnel detection with, 287-290
**netstat command**
application port determination, 284-285
detecting BSD tunnels, 291
detecting Linux tunnels, 291
detecting Solaris tunnels, 292
neighbor determination (Solaris), 287
**Network Address Port Translation-Protocol Translation (NAPT-PT), 438**
**Network Address Translation. *See* NAT**
**Network Address Translation-Protocol Translation (NAT-PT), 437, 459-460**
**network devices**
disabling unnecessary services, 222-224
hardening, overview of, 219
interface hardening, 223-224
LAN switches with FHRP security. *See* FHRPs
limiting router access. *See* router access management
management protocols. *See* management systems
recommendations, list of, 509
security policies for, 505
traffic planes, 220
VTY port access control, 226-229
**Network Foundation Protection, 265**
**network layer, 5. *See also* Layer 3**
**network management**
CiscoWorks LMS, 472
HP OpenView, 472
MTBF, 468
Multi-Router Traffic Grapher, 471
SNMP MIBs for, 469-470
TMN FCAPS model, 467
**Network Mobility (NEMO), 409-410, 503**
**Network Node Manager (NNM), 472**
**Network Time Protocol (NTP), 481-482**
**Next Header (NH) field. *See* NH**
**Next Hop Resolution Protocol (NHRP), 350**
**next-hop addresses, BGP, 99-100**
**NFP (Network Foundation Protection), 265**
**NH (Next Header)**
defined, 16
fragmentation checking issue, 47
next-header numbers, 25-27
**NHRP (Next Hop Resolution Protocol), 350**
**NIQ (Node Information Query)**
filtering, 21
purpose of, 18
reconnaissance with, 62-63
**nmap command, 285**
**NNM (Network Node Manager), 472**
**no ipv6 nd suppress-ra command, 430**
**no ipv6 source-route command, 40**
**no split tunneling, 441**
**Node Information Query. *See* NIQ**
**nodes, 4**
**nonces, 197**
**NSs (Neighbor Solicitations)**
defined, 187
detecting, 201
NUD interval configuration, 256
**NTP (Network Time Protocol), 481-482**
**NUD (Neighbor Unreachability Detection), 255-257**
**Null 0 interfaces, 234**

# O

**object group policy configuration, 168-172**
**Open Shortest Path First version 3. *See* OSPFv3**
**Open Systems Interconnection. *See* OSI model**
**operating system security**
address selection issues, 295-297
application ports, 284
BSD ports, 285
Cisco Security Agent. *See* CSA
dual-stack issues, 281
firewalls. *See* host firewalls
forwarding, stopping, 292-295
ICMPv6 processing, 282-283
ISATAP tunnel detection, 288-289
Linux ports, 284

Microsoft Windows ports, 284
neighbor caches, 285-287
overview of, 281-282
Solaris ports, 285
Teredo tunnel detection, 289-290
tunnel detection, 287-292

**operating systems, IPv6-capable, 6**

**option headers, 379-381**

**OS X known vulnerabilities, 440-441**

**OSI (Open Systems Interconnection) model**
Layer 2. *See* Layer 2
Layer 3, 65-69, 239, 499
Layer 4, 65-69

**OSPFv3 (Open Shortest Path First version 3)**
configuration with IPsec, 248-249
encryption states, 251
IPsec authentication, 247
IPsec interface state, 250
MD5 authentication, 247
MIBs with, 470
neighbor state verification, 249
purpose of, 247
SPI key creation, 247-248
tunnel security issues, 448

**OUIs (Organizational Unique Identifiers), 58**

# P

**P (Provider) routers, 435**

**PAA (PANA Authentication Agent), 409**

**Packet Filter (pf), 304-306**

**packet-flooding attacks, 74-76**

**packets**
defined, 4
filtering. *See* filtering
headers of. *See* headers
spoofing. *See* spoofing

**PAM (Port-to-Application Mapping), 157-158**

**PANA Authentication Agent (PAA), 409**

**parasite6, 190**

**passive-interface command, 240**

**passwords, 225**

**PAT (Port Address Translation), 430**

**Path MTU Discovery (PMTUD), 18, 45, 143, 164**

**payload length field, IPv6 headers, 16**

**PE routers, 108-109**

**peering**
address space issues, 85
black holes, 84
bogons with, 90
disabling route-flap dampening, 104
explicitly configured BGP peers, 92
fast external fallover, 104
filtering on interfaces, 97
graceful restarts, 104
IPsec for, 93
limiting number of prefixes received, 103
link-local peering, 97-102
loopback addresses for, 93
maximizing in BGP, 103-106
prefix filter lists for, 88
resets, 105-106
routers, 84, 92
TTL issues, 94

**performance**
firewall issues, 136
IOS firewall issues, 150
monitoring tools, SNMP-based, 469-472
NetFlow tool, 472-478

**perimeter security**
ACLs for. *See* IOS ACLs
adding IPv6 capabilities, 127
changes, IPv6 versus IPv4, 501-503
Cisco IOS Firewall. *See* IOS firewalls
Cisco PIX/ASA/FWSM firewalls. *See* PIX/ASA/FWSM firewalls
firewalls for. *See* firewalls
overview of, 127-128
policies for, 504
problems with relying on, 127
recommendations, list of, 509
security perimeter model, 127

**perimeter security model, 10**

**physical access security, 224**

**PIM (Protocol Independent Multicast), 340**

**PIX/ASA/FWSM firewalls**
ACLs for, 164-166
ASDM graphical interface, 162
debug command, 174
duplicate address detection, 174-175
EUI-64 address configuration, 159
filtering options, 165
filters versus ACLs, 164

fragmentation security, 172
interface configuration, 159-161
interface example with VLANs, 160
link-local addresses, 159
management access configuration, 161-162
neighbor discovery, 174-177
network topology example, 166
object group policy configuration, 168-172
PMTUD with, 164
reachability confirmation timers, 177
route configuration, 162-163
Router Advertisement (RA) messages, 175
routing tables, viewing, 163
security policy configuration, 164-165, 167-168
show ACLs command, 167
show conn command, 168
show ipv6 interface command, 160
show ipv6 neighbor command, 176-177
show ipv6 routers command, 175
show ipv6 traffic command, 173-174
stateful traffic inspection, 164
stateless autoconfiguration, 159
static unicast addresses, 159
versions, 158

**PMTUD (Path MTU Discovery), 18, 45, 143, 164**
**point-to-point links, 213-215**
**point-to-point tunnels, 422**
**police command, 267**
**policies, security**
ACLs for, 164-166
creating, overview of, 503
extension header, 504
filtering options, 165
filters versus ACLs, 164
hardening hosts and devices, 505
IPsec recommendations, 506
key elements of, 503
LAN threats, 505
management recommendations, 506
object group policy configuration, 168-172
perimeter policies, 504
PIX/ASA/FWSM firewall configuration, 164-168
PMTUD, 164
show ACLs command, 167
show conn command, 168
stateful traffic inspection, 164
transition mechanisms, 506

**pool depletion attacks, 460**
**Port Address Translation (PAT), 430**
**ports**
applications, listening on, 284
BSD, 285
Linux, 284
Microsoft Windows, 284
netstat command for Windows, 284
numbers, changing with IOS firewall, 157
Portproxy, 301
Solaris, 285
VTY, access control, 226-229

**Port-to-Application Mapping (PAM), 157-158**
**prefix delegation threats, 113-119**
**prefix lists, 240-241**
**prefix policies, 295-296**
**privacy address creation, 64**
**privacy extension addresses, 205-208**
**privacy extensions, 484**
**process tables, 263**
**Product Security Incident Response Team (PSIRT), 221**
**protocol headers, 4.** ***See also*** **headers**
**Protocol Independent Multicast (PIM), 340**
**protocol translators**
defined, 417
DNS ALGs, 437
mechanism for, 437-438
NAPT-PT, 438
NAT-PT, 437
proposals to revive, 439

**provider independent address prefixes, 65**
**Provider (P) routers, 435**
**PSIRT (Product Security Incident Response Team), 221**
**published vulnerabilities, vendor list for, 7**
**PVFs (Primary Virtual Forwarder), 260**

# Q

**QoS (Quality of Service)**
defined, 4, 269
ICMP traffic limiting, 271
IP phone priority, 270
MQC commands, 270-271

- scavenger class, 270-273
- settings, viewing, 273-276
- show class-map command, 273
- show policy-map interface command, 273
- syntax changes for IPv6, 270
- ToS bytes, 270
- utility against attacks, 270

# R

**rACLs (Receive ACLs), 265**

**RADIUS (Remote Authentication Dial-In User Service), 111, 230**

**RANCID (Really Awesome New Cisco confIg Differ), 494**

**RAs (Router Advertisements)**
- attack detection, 199-200
- Cisco planned feature set for, 204
- components of, 183
- fake_router6, 185, 200, 202
- flags of, 183
- forged, 185-186
- frequency, setting, 256
- high priority for attack mitigation, 201
- ISATAP use of, 429-430
- latent threats, role in, 440, 461
- lifetimes, 183
- link-layer addresses, 183
- local prefixes, 183
- MTUs, 183
- normal router advertisement mechanism, 183
- NUD with, 256-257
- purpose of, 18
- RA-guard, 203
- rafixd for attack mitigation, 202-203
- reducing attack target scope, 203
- rogue non-malicious, 185
- router priority, 183
- SEND for, 198

**Rbot.AX, 282**

**reachability information vulnerability, 8**

**reachable-time command, 177**

**Really Awesome New Cisco confIg Differ (RANCID), 494**

**recommendations, consolidated list of, 508-511**

**reconnaissance**
- accelerating scanning, 58-59
- alive6 scanning tool, 61
- automation of, 56-58
- defenses against, 63-65
- DNS scans, 56
- multicast, leveraging for, 59-61
- negative effects of, 57
- neighbor cache vulnerability, 62
- NI Query, 62-63
- NMAP for, 57
- OUIs, 58
- overview, 55
- privacy address creation, 64
- registry scans, 56
- router address issue, 64
- scanning overview, 56
- sniffing tool, 61

**redir6 tool, 194**

**redirection issues, 193-195**

**reflection attacks on tunnels, 444**

**reflexive ACLs (rACLs), 147-149**

**registry scans, 56**

**relays**
- 6to4 tunnels, 425-427
- Teredo, 431

**remote access**
- client database details, 364-365
- initiating connections, 364
- IPsec for, overview, 361-362
- ISATAP remote users, 362
- ISATAP routers, 362-363
- ISATAP tunnel status, 363-364
- management access, 226
- pinging remote clients, 367
- remote interfaces, viewing, 365-367
- remote routing tables, viewing, 367
- routers, connecting to, 368
- SSL. *See* SSL VPNs
- topology example, 362
- tunnels. *See* remote-access tunnels

**Remote Authentication Dial-In User Service (RADIUS), 111, 230**

**remote management, 226**

**remote-access tunnels**
- defined, 419
- ISATAP. *See* ISATAP
- static versus dynamic, 362

**Remotely Triggered Black Holes (RTBHs), 84**
**resets of BGP connections, 105-106**
**Return Routability (RR) procedure, 382-385**
**RFC 2460, 4**
**RFC 4029, 107**
**RH0 (Routing Header Type 0)**
- attacks with, 36-39
- CoPP for blocking, 266
- defined, 36
- filtering for MIPv6, 393
- firewall based defenses, 42-43
- policy guidelines for, 504
- preventing attacks, 40-42

**RH2 (Routing Header Type 2)**
- filtering, 392-393
- MIPv6 use of, 379-380

**RIPng (Routing Information Protocol next generation), 241**
**RME (CiscoWorks Resource Manager Essentials), 472, 494**
**RO (Route Optimization), 382**
**rogue AP (access point) problem, 462**
**Route Optimization (RO), 382**
**route-flap dampening, 104**
**router access management**
- access class configuration example, 227
- authentication, 229-230
- console port security, 225
- HTTP access, 230-233
- HTTP policing, 268
- login monitoring, 228-229
- overview, 224
- password encryption, 225
- physical access, 224
- remote access security, 226-228
- SDM access, 231
- SSH, 226-228
- TCP port for HTTP access, 232
- vulnerabilities of management systems, 233

**Router Advertisements. *See* RAs**
**router resource control**
- blocking inbound packets, 263-264
- control plane role in, 262, 265-269
- HTTP policing, 268
- icmp error-interval command, 269
- process table, viewing, 263
- purpose of, 262
- rACLs, 265
- RH0 packet blocking, 266
- show policy-map control-plane command, 267
- SSH policing, 268
- Telnet policing, 268

**Router Solicitations. *See* RSs**
**routers**
- 6to4, 424
- access management. *See* router access management
- ACLs on. *See* ACLs; IOS ACLs
- addresses of, vulnerability from, 64
- BGP. *See* BGP
- buffer overflow vulnerability of, 8
- console ports of, 225
- consumer grade, 111-112
- defined, 4, 240
- DHCP advertisement (RA) messages, 114
- disabling unnecessary services, 222-224
- hop-limit command, 224
- interface hardening, 223-224
- interface monitoring, 468-469
- IOS ACLs. *See* IOS ACLs
- link-layer addresses in RAs, 183
- login monitoring, 228-229
- management systems for. *See* management systems
- mobile access, 378. *See also* MIPv6
- PE routers, 108-109
- peering, 84, 90, 92
- reachability confirmation timers, 177
- resource control. *See* router resource control
- Router Advertisement (RA) messages. *See* RAs
- show ipv6 neighbor command, 176-177
- show ipv6 routers command, 175
- target nature of, 219
- traffic planes, 220
- VTY port access control, 226-229

**routing headers**
- firewall based defenses, 42-43
- preventing RH0 attacks, 40-42
- RH0 attacks, 36-39
- type 0. *See* RH0
- type 2. *See* RH2
- types of, 36
- vulnerabilities overview, 36

**Routing Information Protocol next generation (RIPng), 241**
**routing protocols**
ARP, 239
authentication, 241
EIGRPv6, 242-244
filtering on, 240
IS-IS. *See* IS-IS
limiting hops, 240
NDP. *See* NDP
OSI Layer 3, 239
OSPFv3, 247-255
passive-interface command, 240
prefix lists, 240
purpose of, 240
RIPng, 241
vulnerability overview, 239-241
**routing tables**
change management of, 495
change processing issue, 119
dual stack requirements, 418
remote, viewing, 367
**rpcinfo command, 284-285**
**RR (Return Routability) procedure, 382-385**
**rsmurf6 attack utility, 74-75**
**RSs (Router Solicitations)**
induction by RAs, 183
ISATAP use of, 429
latent threats, role in, 440
purpose of, 18
**RTBHs (Remotely Triggered Black Holes), 84**

# S

**SADBs (Security Association Data Bases), 324**
**Safe Harbor testing, 221**
**SAs (Security Associations)**
defined, 324
IPsec tunnel configuration, 345
SADBs, 324
show crypto ipsec sa command, 346-347
viewing status of, 334
**scanning. *See also* reconnaissance**
DHCPv6 threat from, 210, 213
ISATAP addresses, 455
**scapy6**
forging RAs, 185
fragmented packet test example, 45, 48
Hop-by-Hop extension header test, 30
node information query test example, 63
packet construction with, 30
RH0 attacks, 37-42
router alert packet crafting, 34
spoofing source addresses, 66
**scapy6-teredo, 456**
**scavenger class, 270-273**
**SCP (Secure Copy Protocol), 140, 142**
**SDI (Microsoft Server and Domain Isolation), 327**
**SDM (Security Device Manager), 231**
**SDN (Cisco Self Defending Network), 10-11**
**Secure Copy Protocol (SCP), 140, 142**
**Secure Hash Algorithm version 1. *See* SHA-1**
**Secure Neighbor Discovery. *See* SEND**
**Security Association Data Bases (SADBs), 324**
**Security Associations. *See* SAs**
**security configuration management, 493-495**
**Security Device Manager (SDM), 231**
**Security Information Management Systems (SIMSs), 489**
**security monitoring**
CiscoWorks LMS, 472
configuration management, 493-495
CS-MARS for, 489-493
FCAPS model, 467
forensics for, 483-485
HP OpenView, 472
IDSs for, 485
IPSs for. *See* IPSs
managing IPv6 tunnels, 482-483
Multi-Router Traffic Grapher, 471
NetFlow tool, 472-474, 476-478
overview, 467
router interfaces, 468-469
SNMP MIBs for, 469-470
syslog, 478-481
time, protocol for accurate, 481-482
**security perimeter model, 127. *See also* perimeter security**
**security policies. *See* policies, security**
**Self Defending Network (Cisco SDN), 10-11**

**SEND (SEcure Neighbor Discovery)**
CGA computation, 196-197
DDNS with, 484
deployment issues, 199
IOS implementation of CGA, 198
mechanism of, 196
nonces, 197
purpose of, 19
recommendation for, 505
signature field, 197
vulnerability of, 199
**server multiple network interface issues, 292**
**server provider security**
6VPE, 109-110
address space allocation, 87
attack filtering, 81-82
BGP sessions. *See* BGP
black holes, 84
consumer grade routers, 111-112
customer premises equipment, 110-113
darknets, 84
DDoS mitigation. *See* DDoS
filtering. *See* ingress/egress filtering
iACLs, 97
IPv6 over MPLS, 107-110
large-scale threats. *See* large-scale Internet threats
LSPs using 6PE, 109
multihoming issues, 119-122
overview, 73
peering routers, 92
prefix delegation issues. *See* prefix delegation threats
recommendations, list of, 508
redundant ISPs, 119-120
RFC 4029, 107
tracebacks, 82-83
**service password-encryption command, 225**
**service providers**
lack of for IPv6, 6
security. *See* server provider security
**service-policy command, 266-267**
**services data traffic plane, 220**
**services plane QoS, 269**
**SFTP (Secure FTP), 153-156**
**SHA-1 (Secure Hash Algorithm version 1)**
purpose of, 319
SPI key generation, 248
**shim6, 122**
**show commands**
IOS firewalls, 154
show class-map command, 273
show conn command, 168
show crypto ipsec sa command, 346-347
show flow monitor command, 476
show interfaces tunnel 1 command, 422
show ipv6 access-list command, 167, 469
show ipv6 flow cache command, 475
show ipv6 interface commands, 177, 468
show ipv6 neighbor command, 176-177
show ipv6 routers command, 175
show ipv6 traffic command, 173-174, 468
show ipv6 tunnel command, 422
show logging command, 480
show policy-map control-plane command, 267
show policy-map interface command, 273
**similarities between IPv4 and IPv6, 499-500**
**Simple Network Management Protocol. *See* SNMP**
**SIMSs (Security Information Management Systems), 489**
**site-to-site IPsec configuration**
access lists for, 329
ACLs for, 340-342
active interfaces, viewing, 347-348
all-IPv6 networks, 339-349
central site configuration, 330
encryption options, 349
external interface access lists for, 337
GRE tunnels, 333
IKE configuration, 344-345
IKE parameters, 333
IKE policy creation, 329
IPv6 over IPv4 example, 329-337
IPv6 tunnel status, 343-344
ISAKMP policy, 340
IS-IS issues, 333
multiple sites, dynamic crypto for, 338-339
overview of issues, 328-329
remote site configuration, 331, 342-343
routing tables, viewing, 332, 348-349
SA configuration status, 334, 345
session information, 347
show crypto ipsec sa command, 346-347

state verification, 332
tunnel interface numbers, 340
Unicast RPF, enabling, 337

**site-to-site tunnels, 419**

**SLAAC (Stateless Address Autoconfiguration)**
advertisement mechanism, 183-186
latent threats, role in, 440
prefix delegation threat, 113
privacy extension addresses, 205-208
protection mechanisms, 195

**Slapper worm, 79**

**Smurf attacks, 74-75**

**sniffing, 61, 444**

**SNMP (Simple Network Management Protocol)**
CiscoWorks LMS, 472
community strings, 237
configuration example, 238
disabling, 236
HP OpenView, 472
known attacks, 236
limiting device access, 236
MIBs, 469-470
Multi-Router Traffic Grapher, 471
show command, 237
v3 security features, 238-239
views, 238
vulnerability of, 235

**soft resets, 105**

**Solaris**
address selection issues, 297
firewalls for, 312-313
forwarding, stopping, 294
IPFilter, 313
ndd command, 312-313
neighbor caches, 287
ports, monitoring, 285
tunnel detection, 292

**SONET (Synchronous Optical Network), 213**

**source addresses**
determining legitimacy of, 65
header field for, 16

**Spanning Tree Protocol (STP), 181**

**SPDs (Security Policy Databases)**
IPsec, role in, 323
SAs, 324

**SPIs (security parameter indexes)**
IPv6 tunneling example, 346
Keys, 247
purpose of, 324

**split tunneling, 370**

**spoofing**
antispoofing measures, 128
CEF tables for defense, 66
determining legitimacy of source addresses, 65
ingress/egress filtering to prevent attacks, 66
Layer 3 and Layer 4, 65-69
MIPv6, 388-389
Neighbor Advertisements, 188-189
policy for, 505
redirection messages, 193
Unicast RPF checks, 66-69

**spyware, 282**

**SSH (Secure Shell)**
firewall configuration for, 153-154
policing control protocols, 268
remote access with, 226-228

**SSL (Secure Socket Layer)**
defined, 368
IPsec as replacement, 502
VPNs. *See* SSL VPNs

**SSL VPNs**
address pool usage, 371
advantages of, 368
AnyConnect, 369-373
ASA configuration commands, 370-371
client connection status, 371
clientless style, 368
configuration steps, 369
dual-stack access, 373
remote connections, 369
split tunneling, 370

**stateless autoconfiguration**
defined, 4
Stateless Address Autoconfiguration. *See* SLAAC

**static routing firewall configuration, 162-163**

**static tunnels**
defined, 420
securing, 447-448

**STP (Spanning Tree Protocol), 181**

**Sun Solaris operating system. *See* Solaris**

**SVC (SSL VPN client). *See* AnyConnect**

**SVF (Standby Virtual Forwarders), 260**

**SVGs (Standby Virtual Gateways), 260**

**switches**
ACLs with, 139
Layer 2, importance of, 181-182
**SYN-Flood attacks, 83**
**Synchronous Optical Network (SONET), 213**
**syslog, 478-481**

# T

**TACACS (Terminal Access Controller Access Control System), 229**
**TCP (Transmission Control Protocol)**
ACLs, specifying for, 139
IOS firewall settings for, 151
**telephones, IP, 270**
**Telnet**
policing control protocols, 268
port numbers, changing with IOS firewall, 157
**Teredo**
address assignment, 432
address components, 433
BitTorrent vulnerability, 456
blocking holes drilled by, 457
clients, 431
configuring in Windows, 431-432
consumer grade router issues, 112
CSA for host traffic control, 315
encapsulation method, 430
FPM for blocking, 457-458
IOS, not available on, 431
latent threats, 455, 461
Meeting Space, installed by, 456
Microsoft patches for, 457
NAT devices with, 431
network architecture for, 431
obfuscation requirement, 434
pm-teredo-udp policy, 458-459
policy recommendation, 506
prevalence of, 434
purpose of, 430
registration servers, 431
relays, 431
scapy6-teredo, 456
securing, 455, 457-459
substituting native IPv6 for, 457
tunnel detection, 289-290
UDP packets, blocking, 457
vulnerability issues, 112
**Terminal Access Controller Access Control System (TACACS), 229**
**THC (The Hacker's Choice) IPv6 Attack Toolkit**
alive6, scanning with, 61
dos-new-ipv6, 192
fake_advertise6, 189
fake_router6, 185, 200, 202
forging RAs, 185
parasite6, 190
redir6, 194
sendpees6, 199
Smurf attack tools, 74-75
**threats. *See also* attacks; vulnerabilities of IPv6**
large scale. *See* large-scale Internet threats
latent. *See* latent threats
**time, benefits of accurate, 481-482**
**TLVs (Type Length Values)**
EIGRPv6, 242
IS-IS, 244
**ToS (type of service) bytes, 270**
**tracebacks, 82-83**
**traffic class field, IPv6 headers, 16**
**traffic planes, 220**
**traffic statistics, show command for, 173-174**
**training staff, overview of, 11-12**
**transition mechanisms**
blocking unused, 135
dual-stack approach. *See* dual stacks
latent threats from. *See* latent threats
list of, 417, 462
planning for, 463
protocol translation. *See* protocol translators
purpose of, 417
recommendations, list of, 510
security policies for, 506
tunnel management, 482
tunnels as. *See* tunnels
**Transmission Control Protocol (TCP), 139, 151**
**Transport Layer, 418**
**transport mode, IPsec, 322**
**TTL (Time-to-Live), 94-95**
**TTL attacks, 269**
**tunnels**
6in4. *See* 6in4 tunnels
6to4. *See* 6to4 tunnels
6VPE, 434-436, 459

BSD, detecting unwanted, 291-292
categories of, 419
classification of addresses, 434
configured tunnels, 420-423
defined as transition technique, 417
detecting unwanted, 287-292
dynamic, 420, 449
GRE, 333, 420-421, 447
hacking overview, 444-446
indirect mode mobile, 381
injection attacks, 444-446
inspecting traffic, 134-135
interface numbers for, 340
IOS configuration of 6in4, 421
IPsec tunnel mode, 322, 329-337
IPv4 protocol 41, 134
ipv6ip mode, 421
ISATAP detection, 288-289
ISATAP for. *See* ISATAP
Linux 6in4 configuration, 423
Linux, detecting unwanted, 290-291
LSPs, 435-436
managing, 482-483
Microsoft Windows, detecting unwanted, 287-290
MPLS-VPN, 434-436
NAT interference with, 421
NAT-PT, 437, 459-460
no split tunneling vulnerability, 441
not preferred transition mechanism, 419
perimeter security changes from IPv6, 502
point-to-point, 422
purpose of, 419
recommendations, list of, 510
reflection attacks, 444
remote-access, 419
show interfaces tunnel 1 command, 422
show ipv6 tunnel command, 422
site-to-site, 340, 419
sniffing, 444
Solaris, detecting unwanted, 292
source address checks, 447
split tunneling, 370
static, 420, 447-448
Teredo. *See* Teredo
Unicast RPF checks, 447-448

**Type 2 Routing Header. *See* RH2**

# U

**UDP (User Datagram Protocol)**
ACLs, specifying for, 139
blocking for Teredo, 457
IOS firewall settings for, 152
Teredo tunnels using, 430

**ULA (unique-local addresses, 132**

**unallocated addresses, filtering, 128-133**

**undetermined-transport keyword, 52-54**

**Unicast Reverse Path Forwarding (RPF)**
antispoofing tunnel packets, 447-448
checks, 66-69, 87
enabling for tunnel interfaces, 337
ISATAP checks, 453
policy recommendation for, 504

**unique-local unicast block, 132**

**unreachable messages, ICMP, 223**

**update on IPv6 adoption, 6-7**

**User Datagram Protocol. *See* UDP**

# V

**van Hauser's toolkit. *See* THC (The Hackers Choice) IPv6 Attack Toolkit**

**version field, IPv6 headers, 16**

**versions of IOS, 220-222**

**VFR (Virtual Fragment Reassembly), 49-52, 150**

**Virtual Routing and Forwarding (VRF), 436**

**Vista, Microsoft. *See* Microsoft Windows**

**VLAN microsegmentation, 203**

**VPE (VPN Provider Edge), 434-436, 459**

**VPNs (Virtual Private Networks)**
AnyConnect, 369-373
dynamic multipoint VPNs. *See* DMVPN
IPv6 over MPLS, 107-110
MPLS. *See* MPLS
remote access with. *See* remote access
site-to-site. *See* site-to-site IPsec configuration
SSL. *See* SSL VPNs

**VRF (Virtual Routing and Forwarding), 436**

**VTY port access control, 226-229**

**vulnerabilities of IPv6**
covert channels, 17
extension header vulnerabilities. *See* extension headers

headers, relation to, 17
internal attacks, 8
known, table of, 441
Layer 3 and Layer 4 spoofing, 65-69
multicast overview, 22-23
overview of, 7-8, 15
published, 7
recommendations, list of, 508
reconnaissance issues. *See* reconnaissance

## W

**W32/Sdbot-VJ worm, 80**
**WiFi rogue access points, 462**
**Windows Kernel TCP/IP/IGMPv3 and MLDv2 Vulnerability, 79**
**Windows, Microsoft. *See* Microsoft Windows**
**wireless hotspot latent threat, 461-462**
**Wireshark protocol sniffer, 440**
**worms**
address scanning, 78-79
current IPv6, 79
defined, 77
Morris worm, 74
preventing, 80
propagation of, 78
Slapper worm, 79
W32/Sdbot-VJ worm, 80

## Y

**Yersinia, 182**
HSRP exploitation, 257

## Z

**zombies, 80. *See also* botnets**