



# IPv6 Security Threats and Mitigations



**Neil Lovering**  
**CCIE #1772**  
**Design Consultant**



**Jim Bailey**  
**CCIE #5275**  
**Technical Leader**



June 2, 2008

# Session Objectives

- This session presents IPv6 security in comparison to IPv4 from a threat and mitigation perspective
- Advanced IPv6 security topics like transition options and dual stack environments
- Requirements: basic knowledge of the IPv6 and IPsec protocols as well as IPv4 security best practices

# Agenda

- Types of Threats
- Shared Issues by IPv4 and IPv6
- Specific Issues for IPv6
- IPv6 Security Best Common Practice

# Types of Threats



## A Quick Taxonomy of Threats

# Types of Threats

- **Reconnaissance**—Provide the adversary with information
- **Unauthorized access**—Exploit
- **Header manipulation and fragmentation**—Evade or overwhelm
- **Layer 3–Layer 4 spoofing**—Mask the intent or origin of the traffic
- **ARP and DHCP attacks**—Subvert the host initialization process
- **Broadcast amplification attacks (eg. Smurf, fraggle)**—Amplify the effect of a flood
- **Routing attacks**—Disrupt or redirect traffic flows

## Types of Threats (Cont.)

- **Viruses and worms**—Propagation of the malicious payload
- **Sniffing**—Capturing data
- **Application layer attacks**—Executed at Layer 7
- **Rogue devices**—Unauthorized devices connected to a network
- **Man-in-the-middle (MitM) attacks**—Involve interposing an adversary between two communicating parties
- **Flooding**—Consume enough resources to delay processing of valid traffic

# Shared Issues



## Security Issues Shared by IPv4 and IPv6

# IPv6 Reconnaissance

## Subnet Size Difference

- In IPv4, reconnaissance is relatively easy
  - DNS/IANA crawling (whois) to determine ranges
  - Ping sweeps and port scans
  - Application vulnerability scans
- Default subnets in IPv6 have  $2^{64}$  addresses (/64)
  - At 10 Mpps, it would take more than 50,000 years to scan a single subnet
- NMAP doesn't even support ping sweeps on IPv6 networks
  - Probably because they realize how fruitless it would be

# IPv6 Reconnaissance

## Scanning Methods Are Likely to Change

- Public servers will still need to be DNS reachable
- Increased deployment/reliance on dynamic DNS
  - => More information will be in DNS
- Administrators may adopt easy to remember addresses (::10,::20,::F00D, ::C5C0 or simply IPv4 last octet for dual stack)
- By compromising hosts in a network, an attacker can learn new addresses to scan
- New multicast addresses – all routers (FF05::2) and all DHCP servers (FF05::1:3)
  - No need for reconnaissance anymore

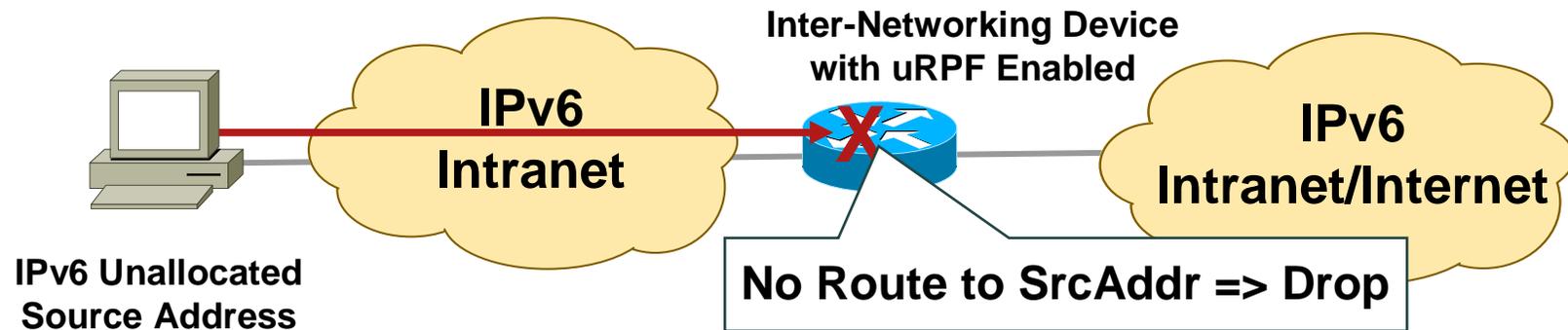
# IPv6 Reconnaissance Best Practices

- **Implement privacy extensions carefully**—for paranoid, use one IPv6 address per connection
- **Filter internal-use** IPv6 addresses at organization border routers—prevent addresses like the all nodes multicast address from becoming conduits for attack
- **Filter unneeded services at the firewall**—just like in IPv4
- **Selectively filter ICMP**—more on this later

See also IETF IPv6 Ops: [draft-ietf-v6ops-scanning-implications-04](#)

# IPv6 Bogon Filtering

- In IPv4, it is easier to block bogons than to permit non-bogons
- In the early days of IPv6, when a small amount of top-level aggregation identifiers (TLAs) were allocated, it was easier to simply block the non-bogons
- Now, IPv6 is in a similar situation as IPv4  
=> Same solution technique = uRPF



## ICMPv4 vs. ICMPv6

- Significant changes
- More relied upon

ICMP Message Type	ICMPv4	ICMPv6
Connectivity Checks	X	X
Informational/Error Messaging	X	X
Fragmentation Needed Notification	X	X
Address Assignment		X
Address Resolution		X
Multicast Group Management		X
Mobile IPv6 Support		X

- => ICMP policy on firewalls needs to change

# IPv6 Header Manipulation

- Unlimited size of header chain (spec wise) can make filtering difficult
- DoS a possibility with poor IPv6 stack implementations
  - More boundary conditions to exploit
  - Can I overrun buffers with a lot of extension headers?

The image shows a packet capture analysis of an IPv6 packet. The packet structure is as follows:

- Frame 1 (423 bytes on wire, 423 bytes captured)
- Raw packet data
- Internet Protocol Version 6
- Hop-by-hop Option Header
- Destination Option Header
- Routing Header, Type 0
- Hop-by-hop Option Header
- Destination Option Header
- Routing Header, Type 0
- Destination Option Header
- Routing Header, Type 0
- Transmission Control Protocol, Src Port: 1024 (1024), Dst Port: bgp (179), Seq: 0, Ack: 0, Len: 51
- Border Gateway Protocol

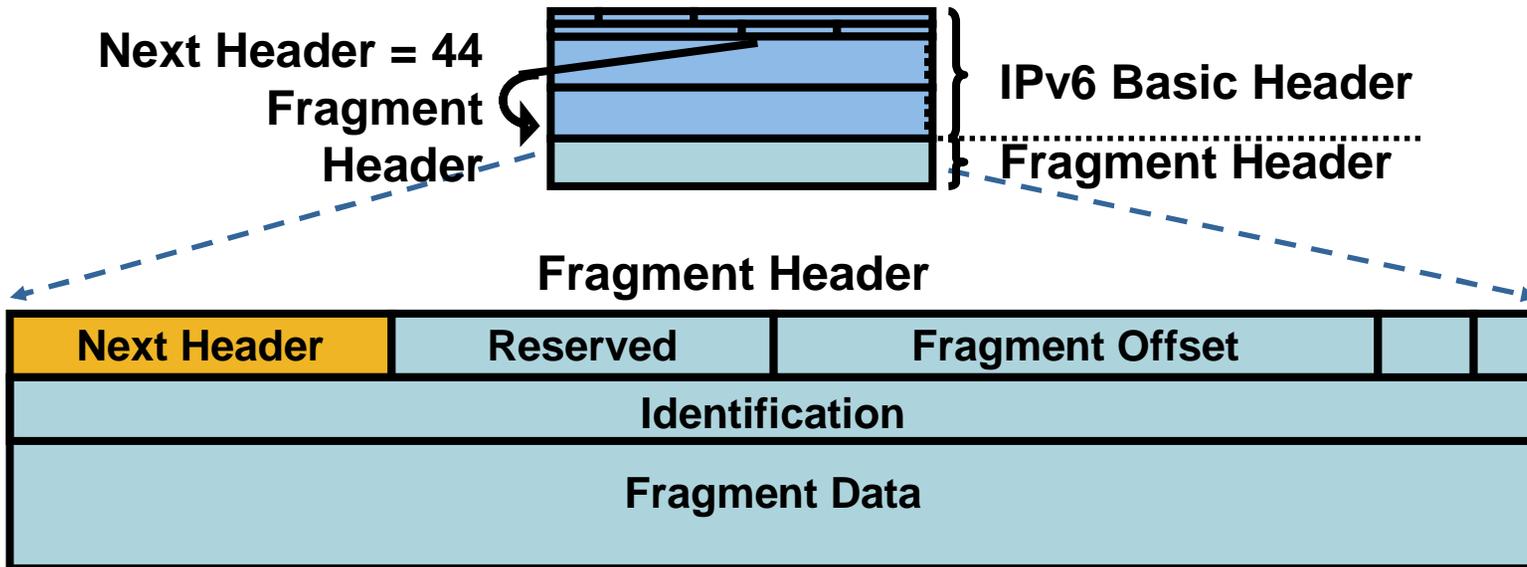
Annotations and rules shown in callout boxes:

- Perfectly Valid IPv6 Packet According to the Sniffer** (points to the entire packet structure)
- Header Should Only Appear Once** (points to the Hop-by-hop Option Header)
- Destination Header Which Should Occur at Most Twice** (points to the Destination Option Headers)
- Destination Options Header Should Be the Last** (points to the final Destination Option Header)

# Fragmentation Used in IPv4 by Attackers

- Fragmentation threats
  - Evasion
  - Insertion
  - Overlap
- Tools like teardrop, whisker, fragrout, etc.
- Makes firewall and network intrusion detection harder
- Used mostly in DoSing hosts, but can be used for attacks that compromise the host

# Fragment Header: IPv6



- In IPv6 fragmentation is done **only** by the end system
- Reassembly done by end system like in IPv4
- Attackers can still fragment in intermediate system on purpose
- ==> a great obfuscation tool

# Header Manipulation and Fragmentation Best Practices

- Deny IPv6 fragments destined to an internetworking device (DoS vector)

Infrastructure ACL

- Ensure adequate IPv6 fragmentation filtering capabilities

For example, drop all packets with the routing header 2 if you do not have MIPv6

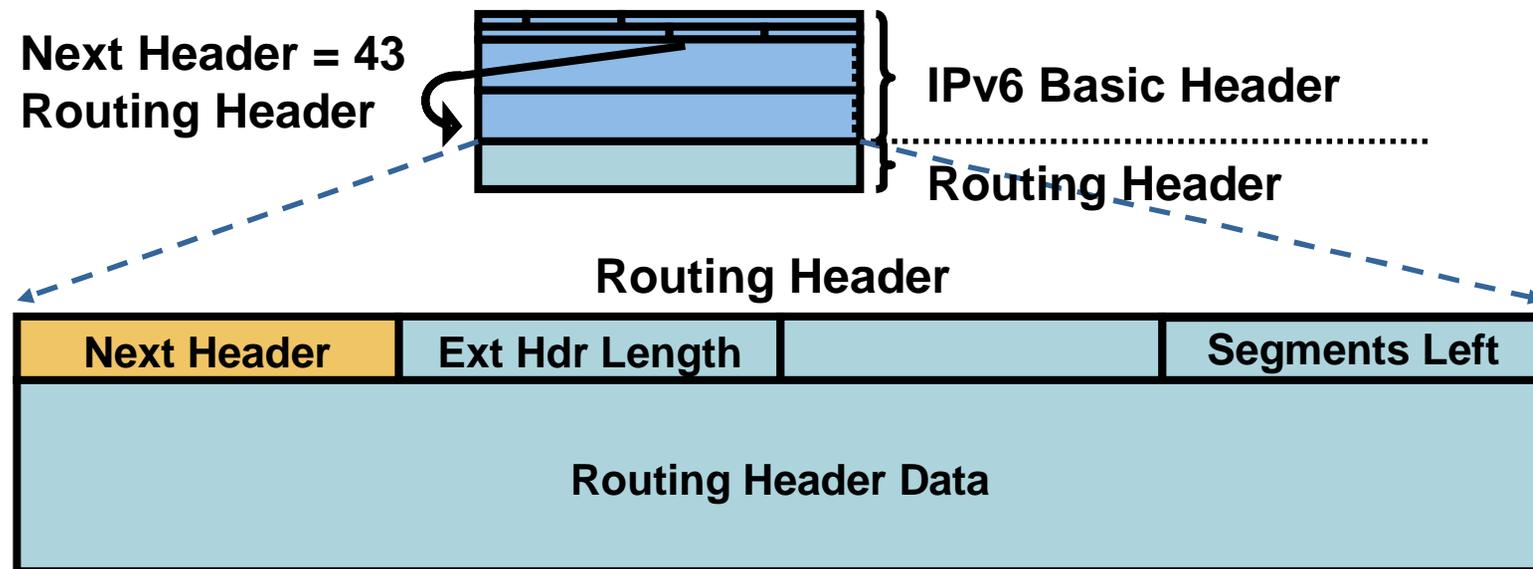
## L3-L4 Spoofing in IPv4

- L4 spoofing can be done in concert with L3 spoofing to attack systems (most commonly running UDP, i.e. SNMP, Syslog, etc.)
- Nearly 50% of the current IPv4 space has not been allocated or is reserved for special use (RFC3330 = ~15%) making it easy to block at network ingress through bogons filtering
- uRPF remains the primary tool for protecting against L3 spoofing
  - If the source address did not come in the proper interface or is unknown – discard

# IPv6 Routing Header

Routing Header Is:

- An extension header
- Processed by the listed intermediate routers
- Similar to IPv4 source routing



## Issues with Routing Header 0

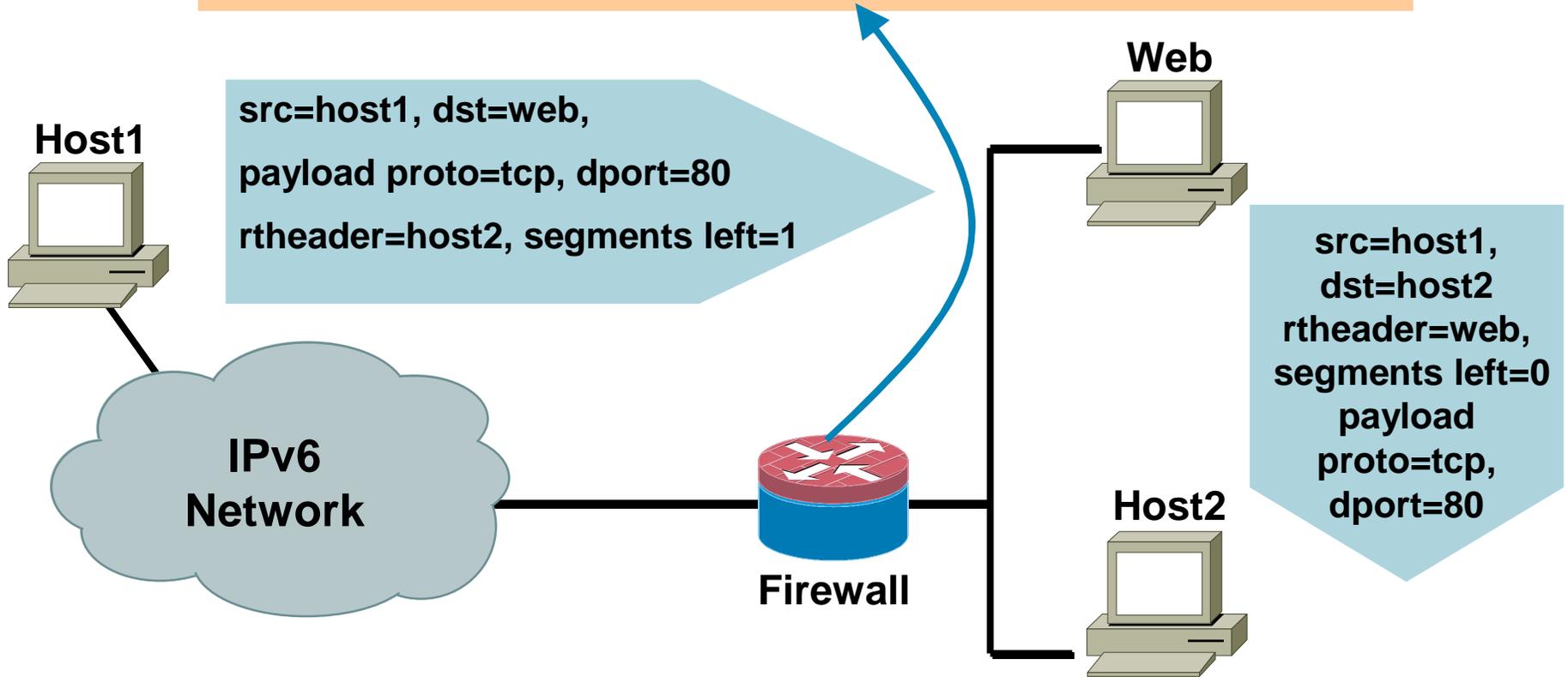
- Could be used as a rebound/relay to the victim
- It is difficult to perform traffic filtering based on destination addresses because destination address is replaced at every routing header processing point
- See draft-savola-ipv6-rh-ha-security-03.txt (expired June 2003)

# Routing Header Issue #1: Traffic Rebound

- Rules on the Firewall

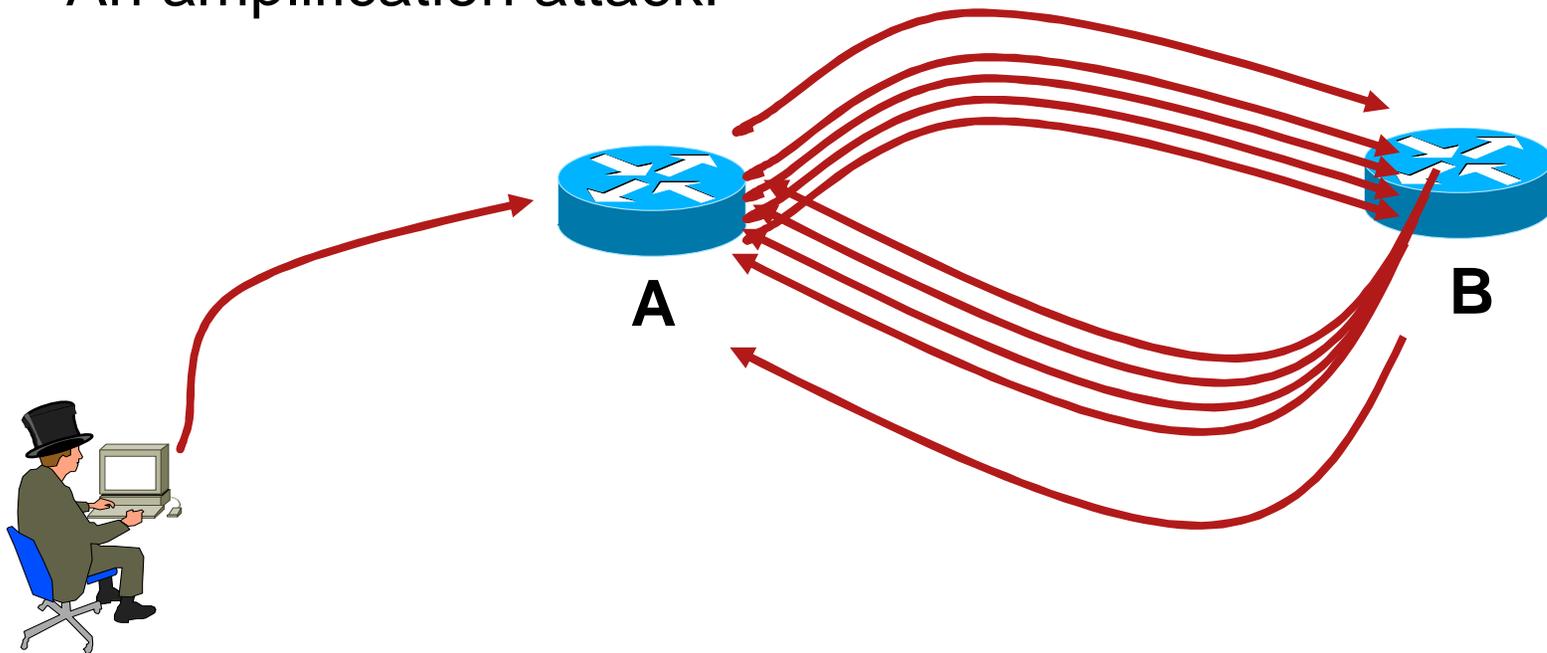
**Permit** protocol tcp from any to webserver on port 80

**Deny** protocol tcp from any to any



# UNCLASSIFIED #2: Routing Header Issue #2: Amplification Attack

- What if attacker sends a packet with RH containing  
A -> B -> A -> B -> A -> B -> A -> B -> A ....
- Packet will loop multiple time on the link R1-R2
- An amplification attack!



# Preventing Routing Header Attacks

- Two types of routing headers
  - Type 0: really evil and mostly useless
  - Type 2: for mobile IPv6 (different discussion)
- Apply same policy for IPv6 as for IPv4:  
Block Routing Header
- At the intermediate nodes

```
no ipv6 source-route
```
- At the edge
  - With an ACL blocking routing header

## “Local Wire” Attacks

- ARP (IPv4 broadcast) is gone, but ...
- IPv6 ICMP-based protocols are insecure
  - Router Solicitation (RS)
  - Router Advertisement (RA)
  - Neighbor Solicitation (NS)
  - Neighbor Advertisement (NA)
- Insecure protocols are subject to MitM attacks
- And with DHCP, the attacker simply needs to put a DHCP server on the wire delivering false information (gateways, DNS servers, etc.)

# IPv6 Neighbor/Router Discovery Attacks in RFC 3756

Sec	Attack name	N/R	R/D	Msgs	1	2	3
4.1.1	NS/NA spoofing	ND	Redir	NA NS	+	+	+
4.1.2	NUD failure	ND	DoS	NA NS	-	+	+
4.1.3	DAD DoS	ND	DoS	NA NS	-	+	+
4.2.1	Malicious router	RD	Redir	RA RS	+	+	R
4.2.2	Default router killed	RD	Redir	RA	+/R	+/R	R
4.2.3	Good router goes bad	RD	Redir	RA RS	R	R	R
4.2.4	Spoofed redirect	RD	Redir	Redir	+	+	R
4.2.5	Bogus on-link prefix	RD	DoS	RA	-	+	R
4.2.6	Bogus address config	RD	DoS	RA	-	+	R
4.2.7	Parameter spoofing	RD	DoS	RA	-	+	R
4.3.1	Replay attacks	All	Redir	All	+	+	+
4.3.2	Remote ND DoS	ND	DoS	NS	+	+	+

1 = Corporate Intranet, 2 = Public Network, 3 = Ad-Hoc Network

- = no concern, + = known solution, R = research

# Secure Neighbor Discovery (SEND) RFC 3971

- Certification paths

Anchored on trusted parties, expected to certify the authority of the routers on some prefixes

- Cryptographically Generated Addresses (CGA – RFC 3972)

IPv6 addresses whose the interface identifier is cryptographically generated

- RSA signature option

Protect all all messages relating to neighbor and router discovery

- Timestamp and nonce options

Prevent replay attacks

# Secure Neighbor Discovery: Caveats

- Private/public key pair on all devices for CGA

- Overhead introduced

Routers have to do many public/private key calculation  
(some may be done in advance of use)

=> Potential DoS target

Routers need to keep more state

- Available: Linux
- Available: Cisco IOS → in 12.5T

# DHCPv6 Threats

- Note: use of DHCP is announced in Router Advertisements
- Rogue devices on the network giving misleading information or consuming resources (DoS)
  - Rogue DHCPv6 client and servers on the network (same threat as IPv4)
  - Rogue DHCPv6 servers on the unique local multicast address (FF05::1:3) (new threat in IPv6)
- Tampering of communication between DHCPv6 relays and servers
- Scanning possible if leased addresses are consecutive

# DHCPv6 Threat Mitigation

- Rogue clients and servers can be mitigated by using the authentication option in DHCPv6

There are not many DHCPv6 client or server implementations using this today

- Cisco Network Registrar

DHCPv6 Server

Leased addresses are random → scanning difficult

Can also lease temporary addresses (like privacy extension)

- For really paranoid: protect the relay to server communications with IPsec (similar to IPv4)

# IPv6 and Broadcasts

- There are no broadcast addresses in IPv6
- Broadcast address functionality is replaced with the appropriate link local multicast address

Link Local All Nodes Multicast—FF02::1

Link Local All Routers Multicast—FF02::2

→ Other than the name, is there really a difference between a “broadcast” and an “all-nodes multicast”?

# Preventing IPv6 Routing Attacks

## Protocol Authentication

- BGP, ISIS, EIGRP no change:
  - An MD5 authentication of the routing update
- OSPFv3 has changed and pulled MD5 authentication from the protocol and instead is supposed to rely on transport mode IPsec
- RIPng also relies on IPsec
- IPv6 routing attack best practices
  - Use traditional authentication mechanisms on BGP and IS-IS
  - Use IPsec to secure protocols such as OSPFv3 and RIPng

# Viruses and Worms in IPv6

- Viruses and email worms: IPv6 brings no change

- Other worms:

IPv4: reliance on network scanning

IPv6: not so easy (see reconnaissance) → will use alternative techniques

- Worm developers will adapt to IPv6
- IPv4 best practices around worm detection and mitigation remain valid
- Potential router CPU attacks if aggressive scanning  
Router will do Neighbor Discovery...

# IPv6 Attacks with Strong IPv4 Similarities

- **Sniffing**

Without IPsec, IPv6 is no more or less likely to fall victim to a sniffing attack than IPv4

- **Application layer attacks**

Even with IPsec, the majority of vulnerabilities on the Internet today are at the application layer, something that IPsec will do nothing to prevent

- **Rogue devices**

Rogue devices will be as easy to insert into an IPv6 network as in IPv4

- **Man-in-the-Middle Attacks (MitM)**

Without IPsec, any attacks utilizing MitM will have the same likelihood in IPv6 as in IPv4

- **Flooding**

Flooding attacks are identical between IPv4 and IPv6

## Specific IPv6 Issues



Issues Applicable only to IPv6

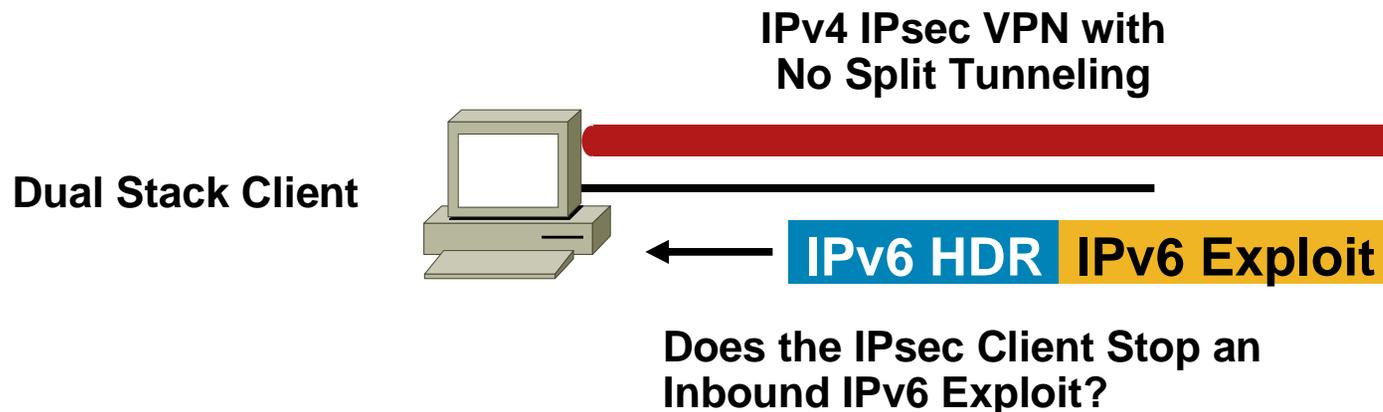
# IPv4 to IPv6 Transition Challenges

- 16+ methods, possibly in combination
  - IP spoofing
- Dual stack
  - Consider security for both protocols
  - Cross v4/v6 abuse
  - Resiliency (shared resources)
- Tunnels
  - Bypass firewalls (protocol 41)

# Dual Stack Host Considerations

- Host security on a dual-stack device
  - Applications can be subject to attack on both IPv6 and IPv4
- Host security controls should block and inspect traffic from both IP versions

Host intrusion prevention, personal firewalls, VPN clients, etc.



# Dual Stack with Enabled IPv6 by Default

- Your host:
  - IPv4 is protected by your favorite personal firewall...
  - IPv6 is enabled by default (Vista, Linux, MacOS, ...)
- Your network:
  - Does not run IPv6
- Your assumption:
  - I'm safe
- Reality:
  - You **may not be** safe (depends on personal firewall capabilities)
  - Attacker sends Router Advertisements
  - Your host configures silently to IPv6
  - You are now under IPv6 attack
- → **Probably time to configure IPv6 on your network**

# IPv6 Tunneling Summary

- RFC 1933/2893 configured and automatic tunnels
  - RFC 2401 IPsec tunnel
  - RFC 2473 IPv6 generic packet tunnel
  - RFC 2529 6over4 tunnel
  - RFC 3056 6to4 tunnel
  - ISATAP tunnel
  - MobileIPv6 (uses RFC2473)
  - Teredo tunnels
- Only allow authorized endpoints to establish tunnels
  - Static tunnels are deemed as “more secure,” but less scalable
  - Automatic tunneling mechanisms are susceptible to packet forgery and DoS attacks
  - These tools have the **same risk** as IPv4, just new avenues of exploitation
  - Automatic IPv6 over IPv4 tunnels could be secured by IPv4 IPsec

# Can We Block Rogue Tunnels?

- Default rogue tunnels by naïve users:
  - Sure, block protocol 41 (6to4) and UDP/3544 (Teredo)
- Really rogue tunnels (covert channels)
  - No way...
  - They will run over a different UDP port of course
- **Use Flexible Packet matching**
  - Blocking all Teredo addresses 2001::/32 in UDP**
- **Deploying native IPv6 (including IPv6 firewalls) is probably a better alternative**

6to4 defined in RFC 3056

Teredo defined in RFC 4380

# IPv6 Security Best Common Practice



# Candidate Best Practices

- Implement privacy extensions carefully
- Filter internal-use IPv6 addresses at the enterprise border routers
- Filter unneeded services at the firewall
- Selectively filter ICMP
- Maintain host and application security
- Determine what extension headers will be allowed through the access control device
- Determine which ICMPv6 messages are required
- Deny IPv6 fragments destined to an internetworking device when possible
- Ensure adequate IPv6 fragmentation filtering capabilities

## Candidate Best Practices (Cont.)

- Implement RFC 2827-like filtering and encourage your ISP to do the same (IP spoofing and DoS attacks)
- Document procedures for last-hop traceback
- Use cryptographic protections where critical
- Use static neighbor entries for critical systems
- Implement ingress filtering of packets with IPv6 multicast source addresses
- Use traditional authentication mechanisms on BGP and IS-IS
- Use IPSec to secure protocols such as OSPFv3 and RIPng
- Use static tunneling rather than dynamic tunneling
- Implement outbound filtering on firewall devices to allow only authorized tunneling endpoints

# Conclusion



# Summary Findings

- IPv6 makes some things better, other things worse, and most things are just different, but no more or less secure

## Better

Automated scanning and worm propagation is harder due to huge subnets

## Worse

Increased complexity in addressing and configuration

Lack of familiarity with IPv6 among operators

Vulnerabilities in transition techniques

- **Most of the legacy issues with IPv4 security remain in IPv6**

For example, ARP security issues in IPv4 are simply replaced with ND security issues in IPv6

## Key Take Away

- Nothing really new in IPv6
- Security enforcement is possible
  - Control your IPv6 traffic as you do for IPv4
- Leverage IPsec to secure IPv6 when possible
- Deploy IPv6, do not wait for a rogue IPv6 network on your infrastructure

# Q and A



UNCLASSIFIED

