

# IPv6 Security



ITU/APNIC/MICT IPv6 Security  
Workshop

23<sup>rd</sup> – 27<sup>th</sup> May 2016

Bangkok

# Before we begin...

---

- Enabling IPv6 on any device means that:
  - The device is accessible by IPv6
  - Interface filters and firewall rules already present in IPv4 **must** be replicated for IPv6
  - Router vty filters already present in IPv4 **must** be replicated for IPv6
- Failure to protect the device after enabling IPv6 means that it is wide open to abuse through IPv6 transport
  - Even though the IPv4 security is in place

# Agenda

---

- ❑ Should I care about IPv6?
- ❑ Issues shared by IPv4 and IPv6
- ❑ Issues specific to IPv6
- ❑ Enforcing a Security Policy in IPv6
- ❑ Secure IPv6 transport over public network
- ❑ IPv6 Security Best Practices

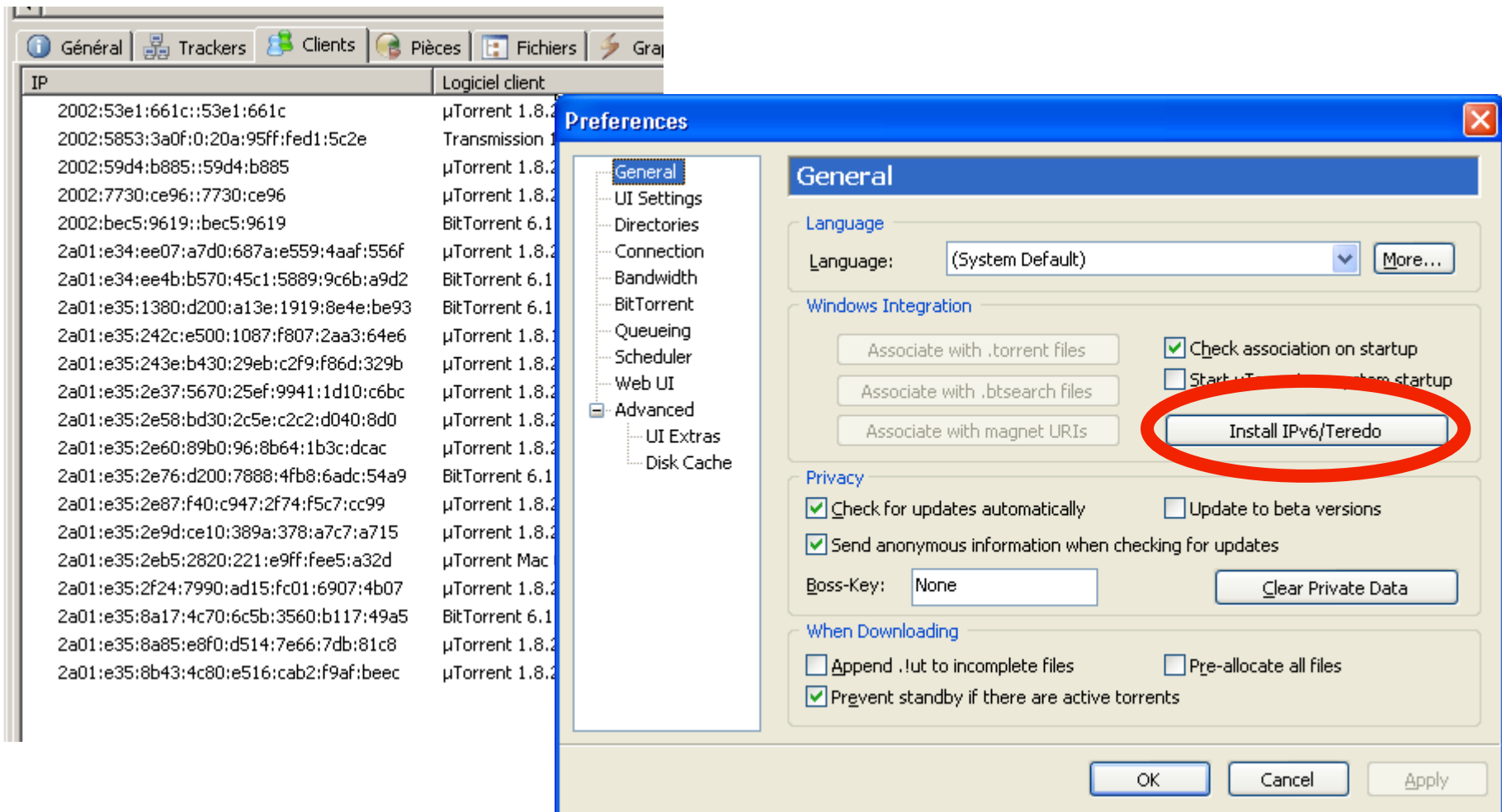
# Should I care?

---

- Is IPv6 in my IPv4 network?
  - Yes!
  - And it is easy to check too
- Look inside IPv4 NetFlow records
  - Protocol 41: IPv6 over IPv4 or 6to4 tunnels
  - IPv4 address: 192.88.99.1 (6to4 anycast server)
  - UDP 3544, the public part of Teredo, yet another tunnel
- Look into DNS requests log for 'ISATAP'

# uTorrent 1.8

- Uses IPv6 by default – released August 2008



# Should I care?

---

- ❑ **Yes, because your end users are already using IPv6**
- ❑ Some transition techniques are aggressive about using IPv6
- ❑ Plus users knowingly configuring IPv6 because “IT” have decided not to supply it by default
  - 6to4 – IPv6 automatic tunnel through IPv4
  - Teredo – tunnel IPv6 through UDP to bypass firewalls and NATs
  - ISATAP – tunnel between IPv6 nodes within organisations
  - GRE or IPv6 in IP tunnels

# Should I care?

---

- Yes, because some operating systems:
  - Have IPv6 turned on by default
    - (most modern OSes)
  - Use IPv6 for administrative communications between devices
    - Windows Server 2008 & 2012, Exchange 2010 etc
- Turning IPv6 off for some of these operating systems actually harms their function and performance
  - Don't do it, even if you think it might be a good idea
- (Yes, this IPv6 deployment by stealth)

# Issues shared by IPv4 and IPv6



Issues facing IPv4 that we can  
find in IPv6...





# Issues shared by IPv4 and IPv6

---

- ❑ Scanning methods
- ❑ Viruses and Worms
- ❑ Filtering
- ❑ Amplification attacks
- ❑ Layer-2 attacks
- ❑ Broadcasts
- ❑ Routing Authentication
- ❑ Hacking

# Scanning

---

- Default subnets in IPv6 have 264 addresses
  - 10 Mpps = more than 50 000 years to scan
  - But different scanning techniques will be used
  - Miscreants will use more intelligent methods for harvesting reachable addresses
- Public servers will still need to be DNS reachable
  - AAAA entries in the DNS
  - More information collected by Google...
  - Network footprint tools like SensePost's Yeti

# Scanning

---

- Administrators usually adopt easy-to-remember addresses
  - Easy to remember:
    - ::10, ::F00D, ::CAFE, ::FADE etc
  - Insert the interface's IPv4 address into the last 32 bits of the interface's IPv6 address:
    - 2001:DB8:10::C0A8:A01 when IPv4 address on interface is 192.168.10.1

# Scanning

---

- Network administrators pick short/simple addresses for infrastructure devices:
  - e.g Loopbacks on 2001:db8::1, 2001:db8::2, etc
- By compromise of hosts in a network
  - Access to one host gives attackers the chance to discover new addresses to scan
- Transition techniques (see later) derive IPv6 address from IPv4 address
  - Plenty of opportunities for more scanning

# Viruses and Worms in IPv6

---

- Viruses & worms
    - No change for IPv6
    - Usual transmission techniques such as IM, email etc are higher up the protocol stack
  - Other worms:
    - IPv4: reliance on network scanning
    - IPv6: not so easy using simple scanning ⇒ will use alternative techniques already discussed
- Worm developers will adapt to IPv6
  - IPv4 best practices around worm detection and mitigation remain valid

# Overloading the CPU

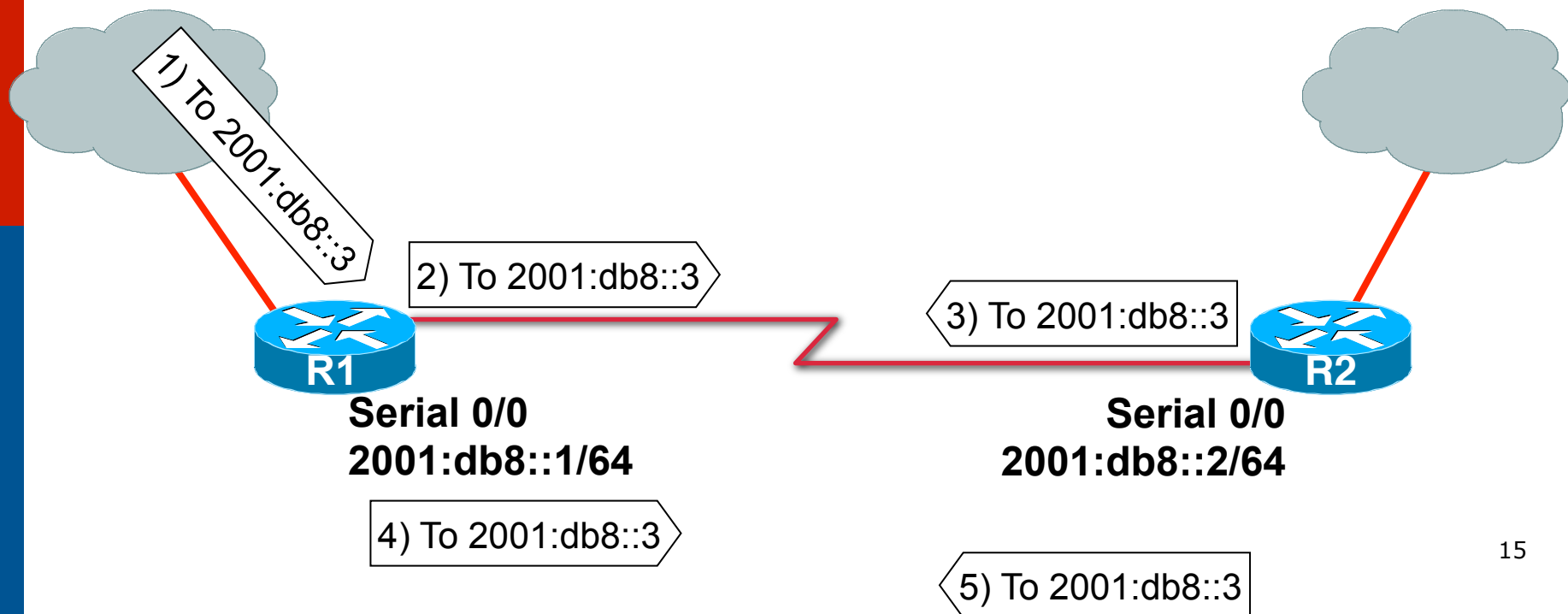
---

- ❑ Aggressive scanning can overload router CPU
  - Router will do Neighbour Discovery, wasting CPU and memory
  - Most routers have built-in rate-limiters which help
- ❑ Using a /64 on point-to-point links  $\Rightarrow$  a lot of addresses to scan!
- ❑ Using infrastructure ACL to prevent this scanning
  - Easy with IPv6 because new addressing scheme can be done 😊

# DoS Example

## Ping-Pong over Physical Point-to-Point

- ❑ Most recent implementations support RFC 4443 so this is not a threat
- ❑ Use of /127 on P2P link recommended (see RFC 6164)
- ❑ Same as in IPv4, on real P2P, "if not for me send it on to the other side", producing looping traffic



# IPv6 Bogon Filtering and Anti-Spoofing

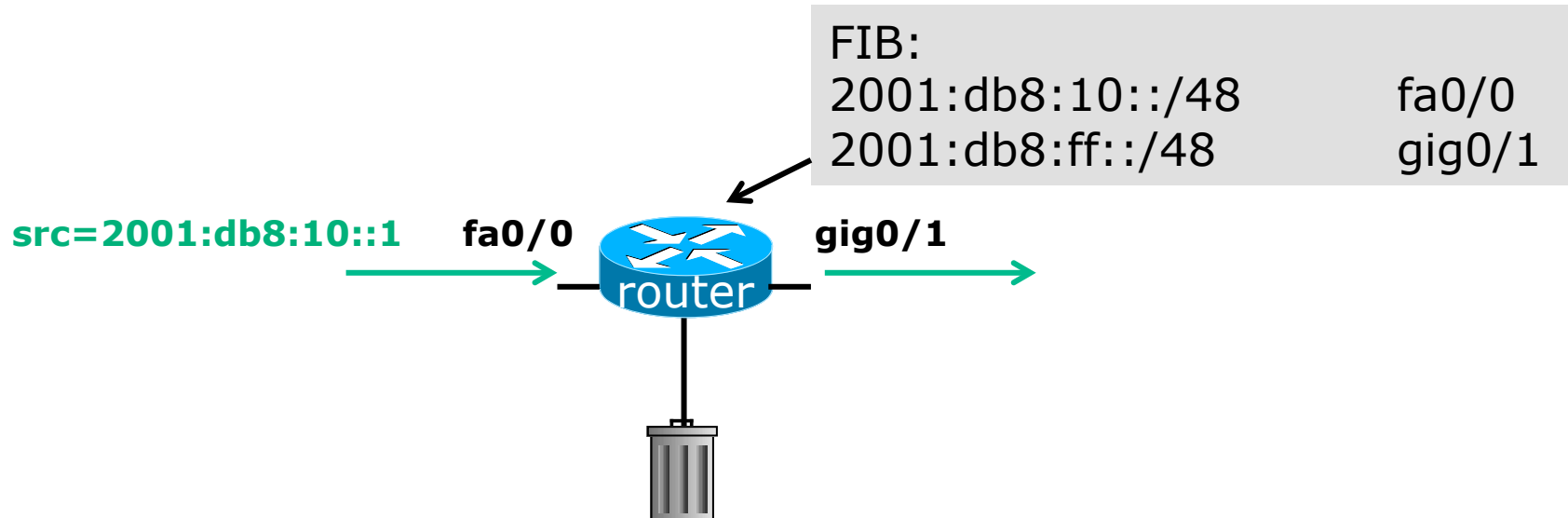
---

- IPv6 has its bogons too:
  - Bogons are prefixes which should not be used or routed on the public Internet
    - <http://www.team-cymru.org/bogon-reference-http.html>
- Similar situation as for IPv4
- BCP 38 still applies!
  - <https://tools.ietf.org/html/bcp38>
- Same technique = uRPF
  - Apply towards all end-users



# Aside: What is uRPF?

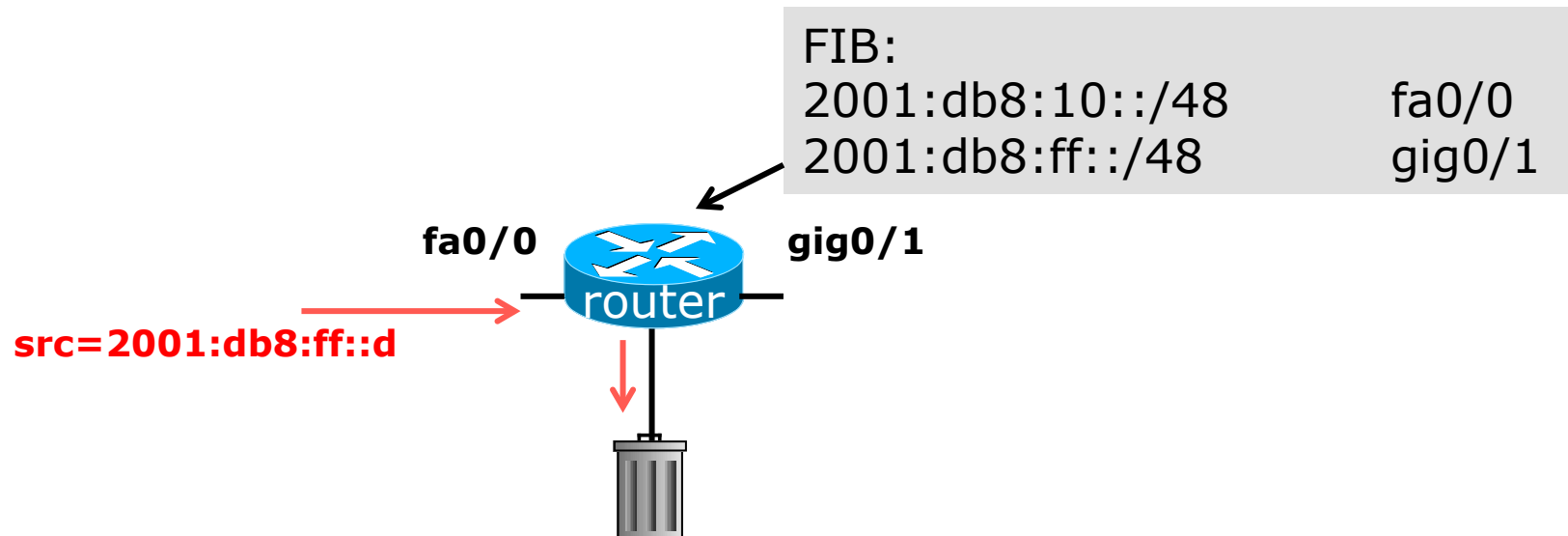
---



- Router compares source address of incoming packet with FIB entry
  - If FIB entry interface matches incoming interface, the packet is forwarded
  - If FIB entry interface does not match incoming interface, the packet is dropped

# Aside: What is uRPF?

---



- Router compares source address of incoming packet with FIB entry
  - If FIB entry interface matches incoming interface, the packet is forwarded
  - If FIB entry interface does not match incoming interface, the packet is dropped

# ICMPv4 vs. ICMPv6

---

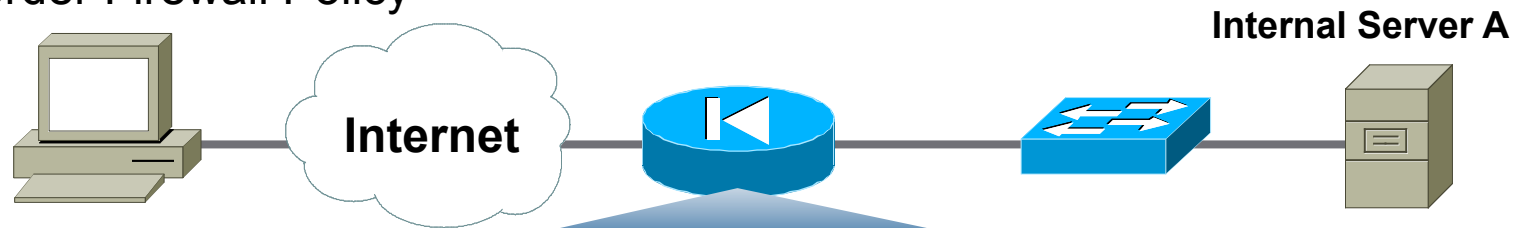
- ❑ Significant changes from IPv4
- ❑ ICMP is relied on much more

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
Router Discovery		X
Multicast Group Management		X
Mobile IPv6 Support		X

- ❑ ICMP policy on firewalls needs fundamental rethink

# Generic ICMPv4

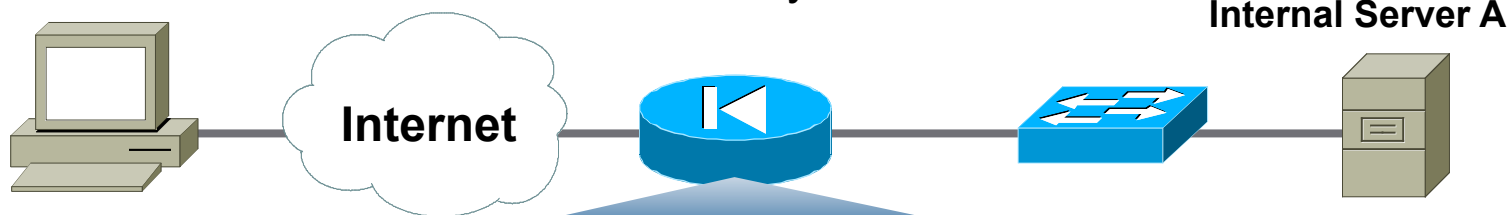
## Border Firewall Policy



Action	Src	Dst	ICMPv4 Type	ICMPv4 Code	Name
Permit	Any	A	0	0	Echo Reply
Permit	Any	A	8	0	Echo Request
Permit	Any	A	3	0	Dst. Unreachable— Net Unreachable
Permit	Any	A	3	4	Dst. Unreachable— Frag. Needed
Permit	Any	A	11	0	Time Exceeded— TTL Exceeded

# Equivalent ICMPv6

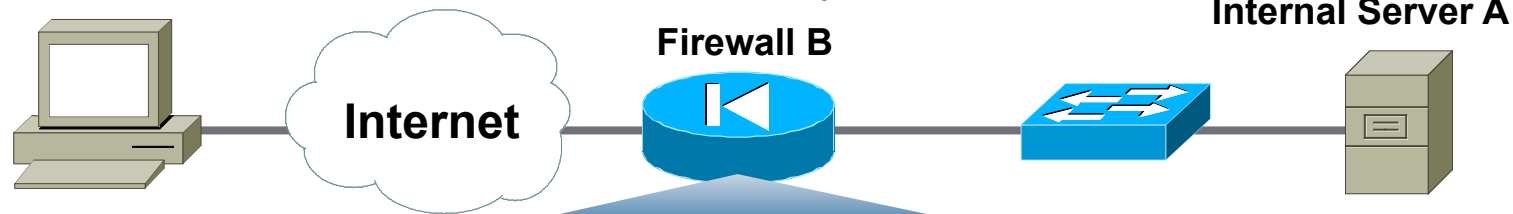
RFC 4890: Border Firewall Transit Policy



Action	Src	Dst	ICMPv6 Type	ICMPv6 Code	Name
Permit	Any	A	128	0	Echo Reply
Permit	Any	A	129	0	Echo Request
Permit	Any	A	1	0	No Route to Dst.
Permit	Any	A	2	0	Packet Too Big
Permit	Any	A	3	0	Time Exceeded— TTL Exceeded
Permit	Any	A	4	0	Parameter Problem

# Equivalent ICMPv6

RFC 4890: Border Firewall Receive Policy

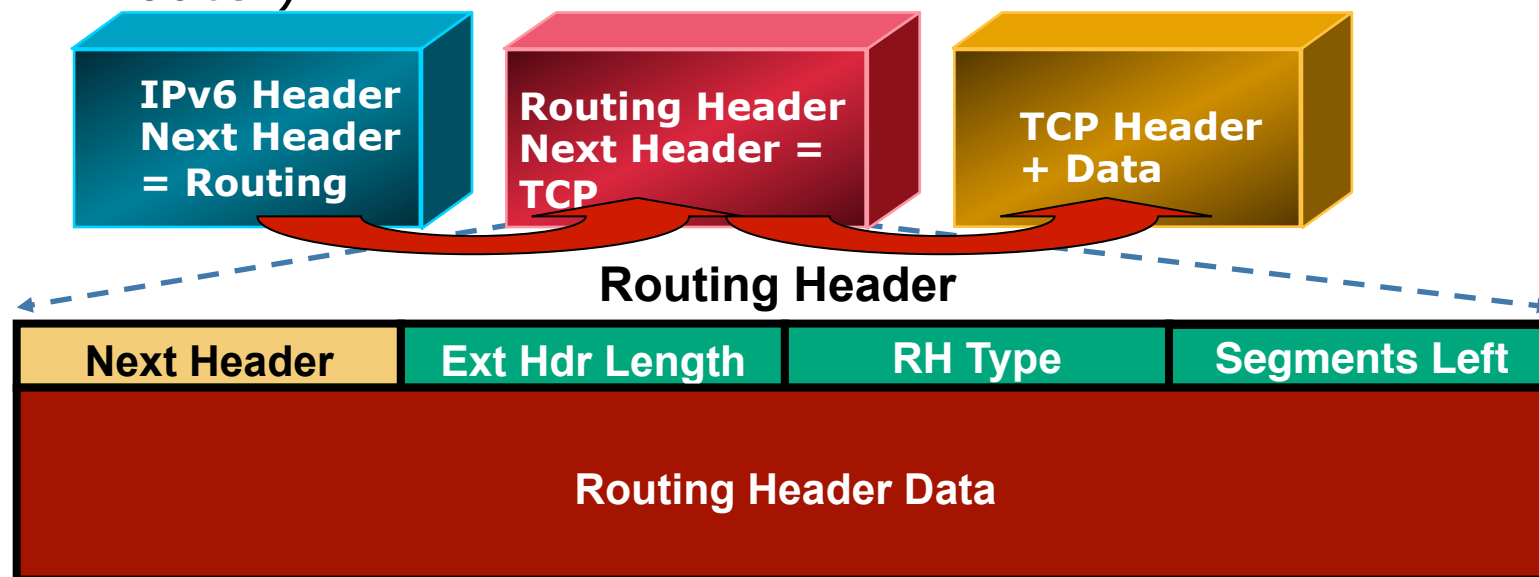


Action	Src	Dst	ICMPv6 Type	ICMPv6 Code	Name
Permit	Any	B	2	0	Packet too Big
Permit	Any	B	4	0	Parameter Problem
Permit	Any	B	130-132	0	Multicast Listener
Permit	Any	B	133/134	0	Neighbor Solicitation and Advertisement
Deny	Any	Any			

For locally generated traffic

# IPv6 Routing Header

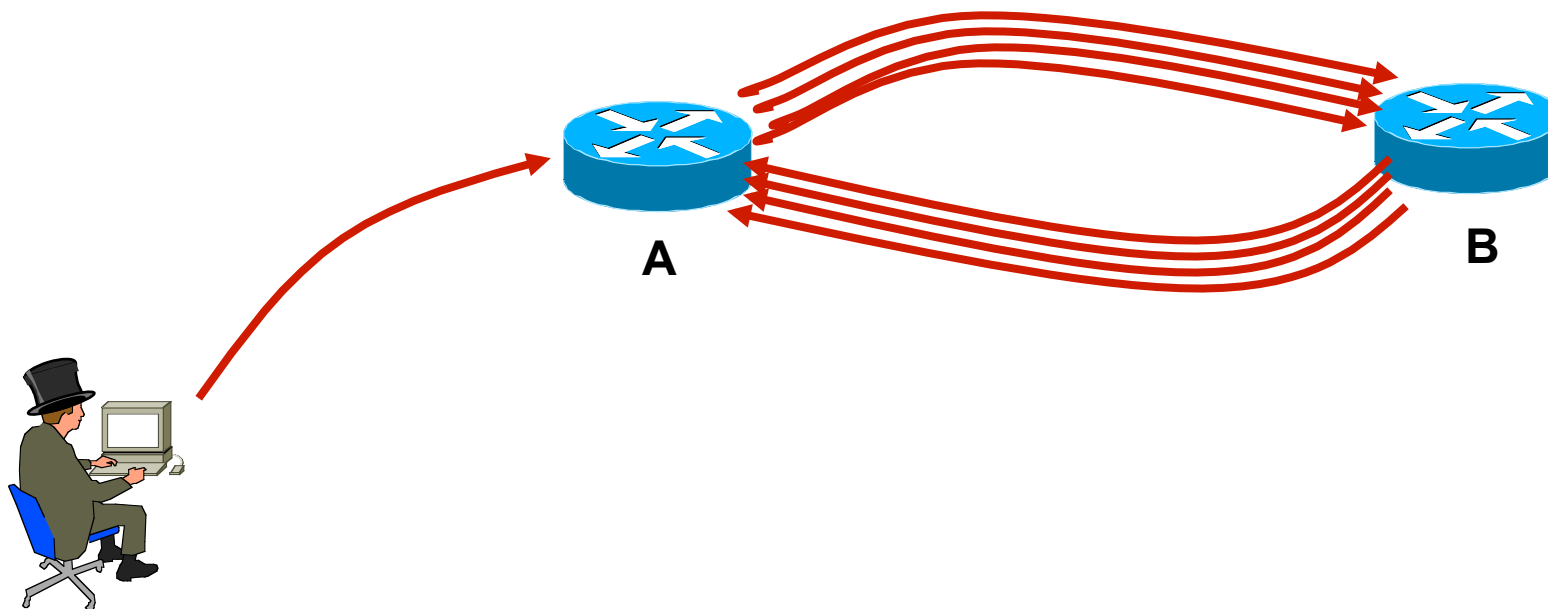
- An extension header
- Processed by the listed intermediate routers
- Two types
  - Type 0: similar to IPv4 source routing (multiple intermediate routers)
  - Type 2: used for mobile IPv6 (single intermediate router)



# Type 0 Routing Header Amplification Attack

---

- ❑ What if attacker sends a packet with a Routing Header containing
  - $A \rightarrow B \rightarrow A \rightarrow B \rightarrow A \rightarrow B \rightarrow A \rightarrow B \rightarrow A \dots$
- ❑ Packet will loop multiple times on the link R1-R2
- ❑ An amplification attack!





# Preventing Routing Header Attacks

---

- ❑ Apply same policy for IPv6 as for IPv4:
  - Block Routing Header type 0
- ❑ Prevent processing at the intermediate nodes
  - `no ipv6 source-route`
    - Windows, Linux, Mac OS: default setting
- ❑ At the edge
  - With an ACL blocking routing header type 0
- ❑ RFC 5095 (Dec 2007) RH0 is deprecated
  - Cisco IOS default changed in 12.4(15)T: no need to type 'no ipv6 source-route'

# Threats on the Layer-2 Link

---

- IPv4 has several threats against layer-2
  - ARP spoofing
  - Rogue DHCP
  - ...
  
- What about IPv6?
  - On WLAN hotspot
  - On ETTx network
  - On hosting service Data Center
  - On ADSL/cable aggregation

# ARP Spoofing is now NDP Spoofing

---

- ❑ ARP is replaced by Neighbour Discovery Protocol
  - Nothing authenticated
  - Static entries overwritten by dynamic ones
- ❑ Stateless Address Autoconfiguration
  - Rogue RA (malicious or not)
  - Node misconfiguration
    - ❑ DoS
    - ❑ Traffic interception (Man In the Middle Attack)
- ❑ Attack tools exist (from THC – The Hacker’s Choice)
  - Parasit6
  - Fakerouter6
  - ...

# ARP Spoofing is now NDP Spoofing

---

- ❑ **BAD NEWS:** nothing like dynamic ARP inspection for IPv6
  - Will require new hardware on some platforms
- ❑ **GOOD NEWS:** Secure Neighbor Discovery (RFC3971)
  - SEND = NDP + crypto
  - But not supported by Windows yet!
  - Crypto means slower...
  - NDPmon toolset (NDP Monitor)
- ❑ **GOOD NEWS:** RA Guard (RFC6105)
  - Superset of SEND
  - Permits RAs based on a set of criteria
- ❑ More **GOOD NEWS:**
  - Private VLAN works with IPv6
  - Port security works with IPv6
  - 802.1X works with IPv6
  - DHCP-PD means no need for NDP-proxy

# IPv6 and Broadcasts

---

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

Link Local All Nodes Multicast	FF02::1
Link Local All Routers Multicast	FF02::2
Link Local All mDNS Multicast	FF02::F

**Anti-spoofing also blocks amplification attacks because a remote attacker cannot masquerade as his victim**

# Preventing IPv6 Routing Attacks: Protocol Authentication

---

- ❑ BGP, ISIS, EIGRP no change:
  - MD5 authentication of the routing update
- ❑ OSPFv3 is different from OSPFv2
  - MD5 authentication dropped from the protocol
  - Authentication relies on transport mode IPsec
- ❑ RIPng and PIM also rely on IPsec
- ❑ IPv6 routing attack prevention best practices
  - Use traditional authentication mechanisms on BGP and IS-IS
  - Use IPsec to secure protocols such as OSPFv3 and RIPng

# OSPFv3 & EIGRP Authentication

---

## □ OSPFv3:

```
ipv6 router ospf 30
  area 0 authentication ipsec spi 256 md5
    1234567890ABCDEF1234567890ABCDEF
```

## □ EIGRP:

```
interface Ethernet0/0
  ipv6 authentication mode eigrp 100 md5
  ipv6 authentication key-chain eigrp 100 MYCHAIN
  !
key chain MYCHAIN
  key 1
    key-string my-eigrp-pw
```

# BGP and ISIS Authentication

---

## □ BGP:

```
router bgp 10
  address-family ipv6
    neighbor 2001:db8::4 remote-as 11
    neighbor 2001:db8::4 password bgp-as11-pw
```

## □ ISIS:

```
interface Serial0/0
  isis authentication mode md5
  isis authentication key-chain MYCHAIN
!
key chain MYCHAIN
  key 1
    key-string my-isis-pw
```



# IPv6 Attacks with Strong IPv4 Similarities

---

- **Sniffing**
  - Without IPSec, IPv6 is as vulnerable to sniffing as IPv4
- **Application layer attacks**
  - 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 strong mutual authentication, any attacks utilizing MITM will have the same likelihood in IPv6 as in IPv4
- **Flooding**
  - Flooding attacks are identical between IPv4 and IPv6

# By the Way: It Is Real ☹️

## IPv6 Hacking/Lab Tools

---

- ❑ Sniffers/packet capture
  - Snort
  - TCPdump
  - Sun Solaris snoop
  - COLD
  - Wireshark
  - Analyzer
  - Windump
  - WinPcap
- ❑ DoS Tools
  - 6tunneldos
  - 4to6ddos
  - Imps6-tools
- ❑ Scanners
  - IPv6 security scanner
  - Halfscan6
  - Nmap
  - Strobe
  - Netcat
- ❑ Packet forgers
  - Scapy6
  - SendIP
  - Packit
  - Spak6
- ❑ Complete toolkit
  - <https://www.thc.org/thc-ipv6/>

# Specific IPv6 issues



New features in IPv6 introduce  
new problems...

# Specific IPv6 Issues

---

- ❑ IPv6 header manipulation
- ❑ Link Local vs Global Addressing
- ❑ Transition Challenges
- ❑ 6to4, 6VPE
- ❑ v4/v6 translation issues
- ❑ IPv6 stack issues

# IPv6 Header Manipulation

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

The image shows a Wireshark packet capture 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 red circles highlight the following issues:

- Header Should Only Appear Once:** Points to the first Hop-by-hop Option Header and the first Destination Option Header.
- Destination Header Which Should Occur at Most Twice:** Points to the second Destination Option Header.
- Destination Options Header Should Be the Last:** Points to the third Destination Option Header.

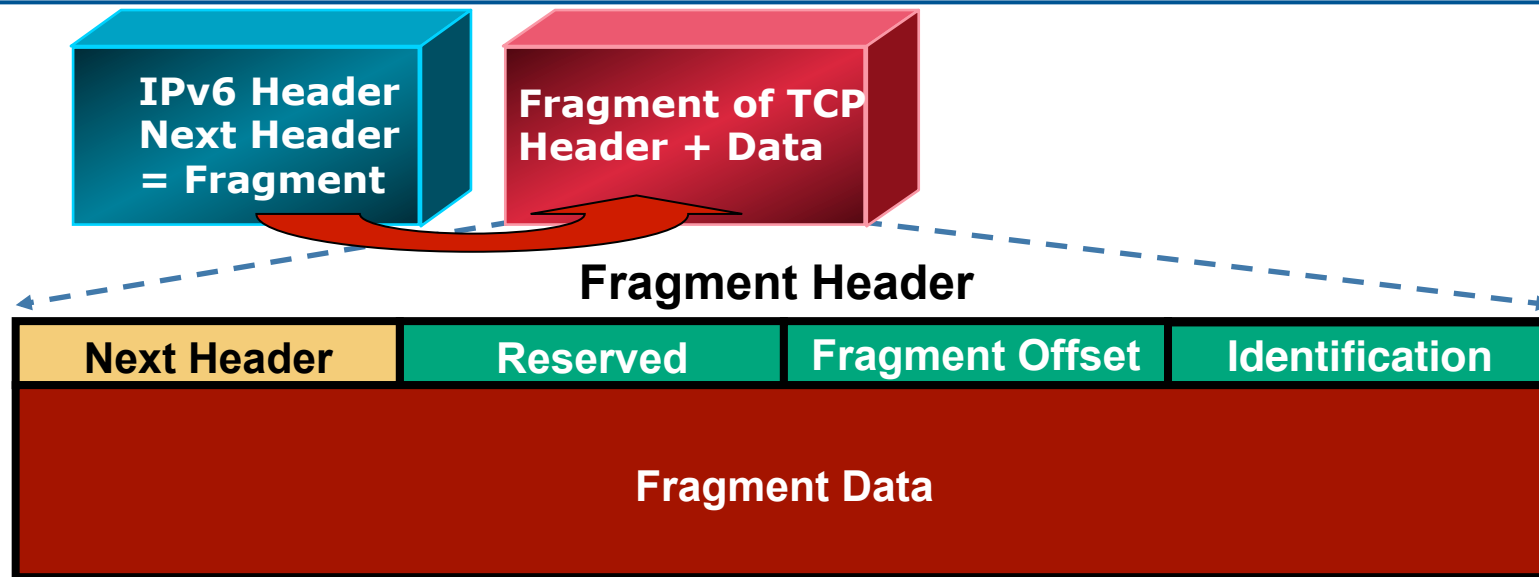
**Perfectly Valid IPv6 Packet According to the Sniffer** is noted for the entire packet structure.

# Parsing the Extension Header Chain

- Finding the layer 4 information is not trivial in IPv6
  - Skip all known extension header
  - Until either known layer 4 header found ⇒ **SUCCESS**
  - Or unknown extension header/layer 4 header found... ⇒ **FAILURE**



# Fragment Header: IPv6



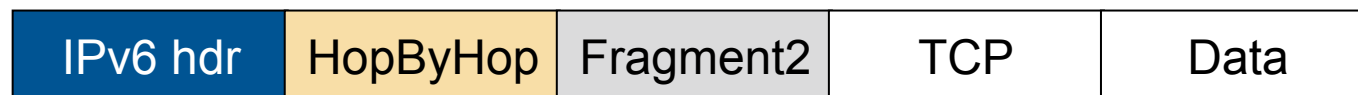
- ❑ According to the IPv6 RFC, fragmentation is only done by the end system
  - But in some cases, routers act as an end system
- ❑ Reassembly done by end system like in IPv4
- ❑ Attackers can still cause fragmentation in end/intermediate systems
  - A great obfuscation tool to hide attacks on IPS & firewall

# Parsing the Extension Header Chain

## Fragmentation Matters!

---

- ❑ Extension headers chain can be so large that the header itself is fragmented!
- ❑ Finding the layer 4 information is not trivial in IPv6
  - Skip all known extension headers
  - Until either known layer 4 header found ⇒ **SUCCESS**
  - Or unknown extension header/layer 4 header found ⇒ **FAILURE**
  - Or end of extension headers ⇒ **FAILURE**



Layer 4 header is  
in 2<sup>nd</sup> fragment



# IPv6 Fragments

---

- ❑ Unlimited size of the extension header chain is a source of potential problems
- ❑ Recommendation is to block all IPv6 fragments on perimeter filters:
  - E.g. for Cisco IOS:

```
ipv6 access-list border-acl-in
...
deny ipv6 any any fragments
...
```

# Link-Local vs. Global Addresses

---

- ❑ Link-Local addresses (FE80::/10) are isolated
  - Cannot reach outside of the link
  - **Cannot be reached from outside of the link 😊**
- ❑ Could be used on the infrastructure interfaces
  - Routing protocols (including BGP) work with LLA
  - Benefit: no remote attack against your infrastructure
    - ❑ Implicit infrastructure ACL
  - Note: need to provision loopback for ICMP generation
  - LLA can be configured statically (not the EUI-64 default) to avoid changing neighbour statements when changing MAC

# IPv6 Transition Technologies Security



From IPv4 to IPv6, securely

# Actively deployed Transition Technologies

---

- Dual stack
- Generic Tunnels
- 6to4
- ISATAP
- Teredo
- NAT64 (and NAT)
- 6rd
- DS-Lite
- 464XLAT
- 6PE & 6VPE

# IPv4 to IPv6 Transition Challenges

---

- ❑ Many competing methods, several may be deployed at the same time
- ❑ Dual stack
  - Consider security for both protocols
  - Cross v4/v6 abuse
  - Resiliency (shared resources)
- ❑ Tunnels
  - Bypass firewalls (protocol 41 or UDP)
  - Bypass other inspection systems
  - Render Netflow blind
  - Traffic engineering becomes tough
  - Asymmetrical flows (6to4)

# Dual Stack with IPv6 on by Default

---

- Your host:
  - IPv4 is protected by your favorite personal firewall...
  - IPv6 is enabled by default (Windows, Linux, Mac OS/X, ...)
- Your network:
  - Does not run IPv6
- Your assumption:
  - I'm safe
- Reality
  - You are not safe
  - Attacker sends Router Advertisements
  - Your host silently configures IPv6
  - You are now under IPv6 attack
- ⇒ Probably time to think about IPv6 in your network

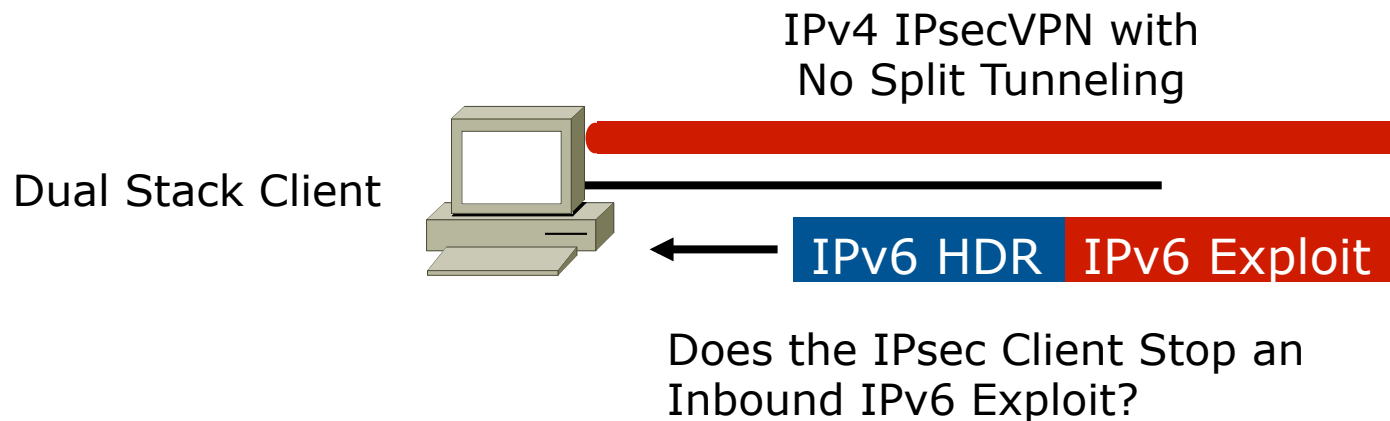
# Dual Stack Host Considerations

---

- Host security on a dual-stack device
  - Applications can be subject to attack on both IPv6 and IPv4
  - **Fate sharing: connectivity is as secure as the least secure stack...**
- Host security controls **must** filter and inspect traffic from both IP versions
  - Host intrusion prevention, personal firewalls, VPN clients, etc.

# Split Tunnelling on VPNs

- VPNs are especially vulnerable:
  - Split tunneling
    - IPv4 traffic goes over the IPSEC Tunnel, but
    - IPv6 traffic goes native, and is potentially vulnerable
  - IPv6 host is vulnerable to incoming exploits





# How to block Rogue Tunnels?

---

- ❑ Rogue tunnels by naïve users:
  - Sure, block IP protocol 41 and UDP/3544
  - In Windows:

```
netsh interface 6to4 set state state=disabled undoonstop=disabled
netsh interface isatap set state state=disabled
netsh interface teredo set state type=disabled
```

- ❑ Really rogue tunnels (covert channels)
  - No easy way...
  - Teredo will run over a different UDP port of course
- ❑ Deploying native IPv6 (including IPv6 firewalls and IPS) is best/easier alternative
- ❑ Or disable IPv6 (uh?)

# 6to4 Issues

---

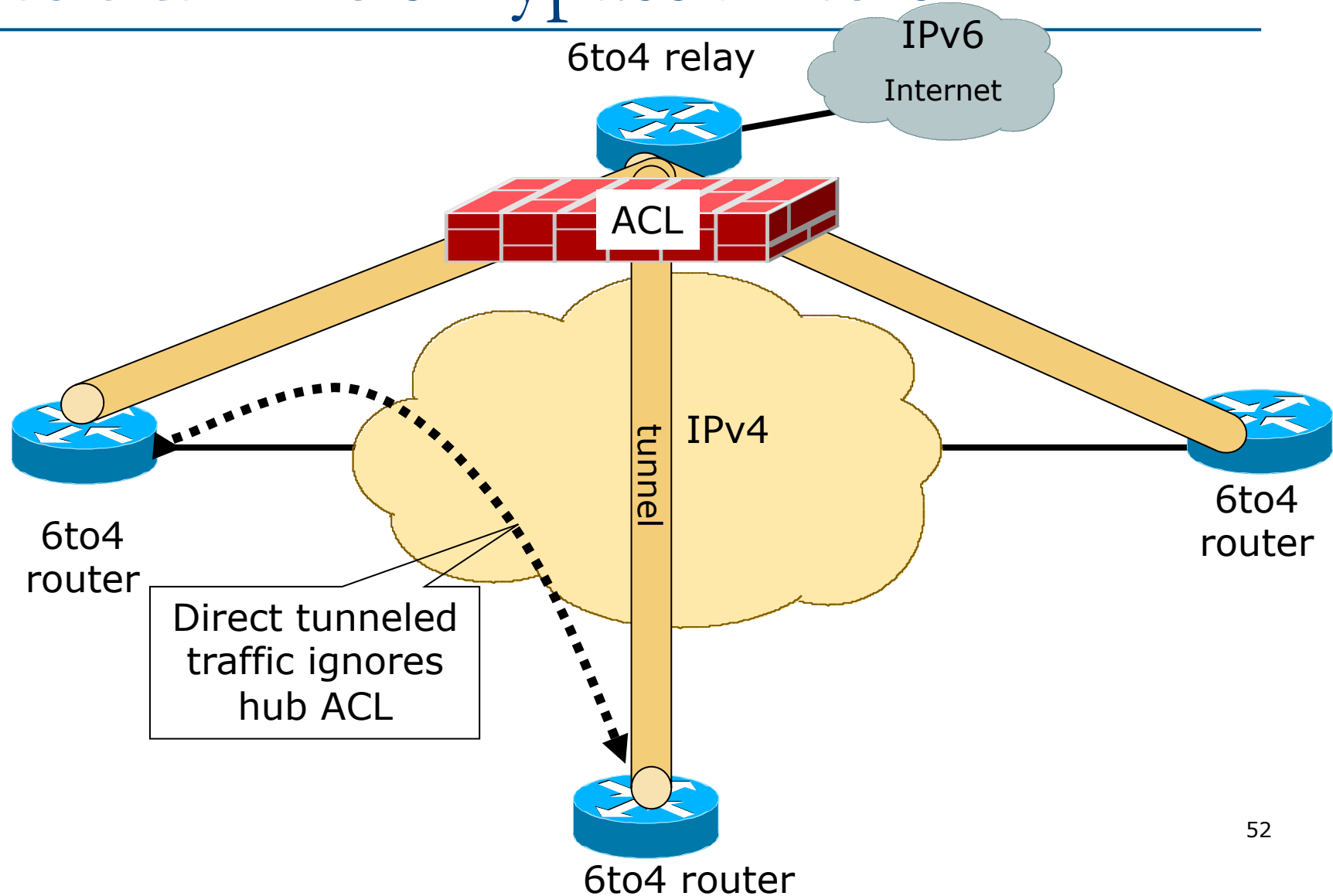
- ❑ Automatic tunnelling technology
- ❑ Bypasses filters, firewalls, most intrusion detection systems
- ❑ Two components:
  - 6to4 client
  - 6to4 relay
- ❑ 6to4 host might be IPv4 protected – what about IPv6 protection, filters,...?
- ❑ 6to4 relay
  - 6to4 host picks topologically closest relay
  - Outbound traffic – your ISP's relay
  - Return traffic – whose relay??

# 6to4 Tunnels Bypass Filters

---

- 6to4 tunnel to another 6to4 host on local network
  - Results in IPv6 packets going from one IPv6 host to another IPv6 over IPv4
  - Bypasses IPv6 packet filters on central host
  - Bypasses IPv4 packet filters on central host
  - ⇒ Major security risk

# 6to4 Tunnels Bypass Filters



# 6to4 Relay Security Issues

---

- Traffic injection & IPv6 spoofing
  - Prevent spoofing by applying uRPF check
  - Drop 6to4 packets whose addresses are built on IPv4 bogons
    - Loopback
    - RFC 1918
- Redirection and DoS
  - Block most of the ICMPv6 traffic:
    - No Neighbor Discovery
    - No link-local traffic
    - No redirect

# 6to4 Relay Security Issues

---

- Traffic is asymmetric
  - 6to4 client/router → 6to4 relay → IPv6 server:
    - Client IPv4 routing selects the relay
  - IPv6 server → 6to4 relay → 6to4 client/router:
    - Server IPv6 routing selects the relay
  - Cannot insert a stateful device (firewall, ...) on any path
- Potential amplification attack (looping IPv6 packet) between ISATAP server & 6to4 relay
  - Where to route: 2002:isatap::/48 ?
  - Where to route: isatap\_prefix::200:5efe:6to4?

# ISATAP issues

---

- Intra-site tunneling protocol
  - Designed to let isolated IPv6 clients speak to other isolated IPv6 enabled devices over a site's IPv4 infrastructure
- Security considerations:
  - Client IPv6 filtering/firewalling?
  - Tunnel technology could bypass inter-departmental controls used for IPv4
  - Who runs the domain's ISATAP server?

# Teredo Issues

---

- ❑ UDP based tunnelling technology to allow remote IPv6 clients connect to IPv6 Internet over IPv4 infrastructure
  - Uses UDP
  - Bypasses firewalls and traverses NATs
- ❑ Already seen the “bittorrent” case at the start of the presentation
- ❑ Severe security risk for any organisation
  - Client IPv6 filters?
  - Firewall bypass
  - Who runs the remote Teredo relay?
  - Runs on non-default UDP ports too



# Translation Issues

---

- Whether NAT64 or NAT444
- Shared IPv4 address among different subscribers
  - Per-IP address reputation means that bad behaviour by one affects multiple subscribers
  - Sending ICMP Packet-too-big to common server means bandwidth reduction for all subscribers sharing that source IP address
  - Huge amount of log traffic for Lawful Intercept (but there are other ways to keep track)

## 6rd Issues

---

- Based on 6to4, so potentially inherits most of 6to4's security considerations
  - Securing IPv6 traffic on 6rd client in the same way as for native IPv4 traffic
- 6rd-relay is controlled by ISP though
  - Avoids "publicly operated" relay problem which plagues 6to4

# DS-Lite & 464XLAT Issues

---

- ISP has native IPv6 backbone
  - And no IPv4
- IPv4 tunnelled through IPv6
- CPE is dual stack towards the end user
  - Usual dual stack security considerations
- ISP core tunnel termination (Large Scale NAT)
  - Faces all the security and scaling considerations that any NAT device would face

# 6VPE Security Issues

---

- ❑ 6PE (dual stack without VPN) is a simple case
- ❑ Security is identical to IPv4 MPLS-VPN, see RFC 4381
- ❑ Security depends on correct operation and implementation
  - QoS prevent flooding attack from one VPN to another one
  - PE routers must be secured: AAA, iACL, CoPP ...

# 6 VPE Security Issues

---

- MPLS backbones can be more secure than “normal” IP backbones
  - Core not accessible from outside
  - Separate control and data planes
- PE security
  - Advantage: Only PE-CE interfaces accessible from outside
  - Makes security easier than in “normal” networks
  - IPv6 advantage: PE-CE interfaces can use link-local for routing
  - ⇒ completely unreachable from remote (better than IPv4)

# IPv6 Security Policies



So how do we go about  
securing the network...?

# IPv6 Security Policy

---

- Access control lists
  - Configuration
  - Implicit Rules
- Interface and VTY filtering
- IPv6 NetFlow
- Enterprise Security

# Cisco IOS IPv6 Extended Access Control Lists

---

- **Very much like in IPv4**
  - Filter traffic based on
    - Source and destination addresses
    - Next header presence
    - Layer 4 information
  - Implicit deny all at the end of ACL
  - Empty ACL means traffic allowed
  - Reflexive and time based ACL
- Known extension headers (HbH, AH, RH, MH, destination, fragment) are scanned until:
  - Layer 4 header found
  - Unknown extension header is found



# IPv6 ACL Implicit Rules

## RFC 4890

---

- Implicit entries exist at the end of each IPv6 ACL to allow neighbour discovery:

```
permit icmp any any nd-na  
permit icmp any any nd-ns  
deny ipv6 any any
```

# IPv6 ACL Implicit Rules:

## Adding a deny-log

---

- ❑ The IPv6 beginner's mistake is to add a 'deny log' at the end of the IPv6 ACL

```
. . .  
! Now log all denied packets  
deny IPv6 any any log  
! Oooops . . . I forget about these implicit lines  
permit icmp any any nd-na  
permit icmp any any nd-ns  
deny ipv6 any any
```

- ❑ Instead, explicitly add the implicit ACL

```
. . .  
! Now log all denied packets  
permit icmp any any nd-na  
permit icmp any any nd-ns  
deny ipv6 any any log
```

# To filter ICMPv6 or not?

---

- ❑ Many administrators are very accustomed to severely filtering ICMPv4
  - Due to history – the ICMP DoS attacks from the late 90s and early 2000s.
  - Blocking all ICMPv4 doesn't really hurt IPv4 too much
    - ❑ Stops Path MTU Discovery
    - ❑ Makes troubleshooting incredibly hard
- ❑ Severely filtering ICMPv6 will cause serious harm to IPv6, or even preventing IPv6 from working
  - RFC4890 filtering *or*
  - Rate-limit ICMPv6 and allow it all

# Example: RFC 4890 ICMP ACL

---

```
ipv6 access-list RFC4890
  permit icmp any any echo-reply
  permit icmp any any echo-request
  permit icmp any any 1 3
  permit icmp any any 1 4
  permit icmp any any packet-too-big
  permit icmp any any time-exceeded
  permit icmp any any parameter-problem
  permit icmp any any mld-query
  permit icmp any any mld-reduction
  permit icmp any any mld-report
  permit icmp any any nd-na
  permit icmp any any nd-ns
  permit icmp any any router-solicitation
```

# Example: Rogue RA & DHCP ACL

---

- If rogue RA or rogue DHCP server detected on network, how to deal with it?

```
ipv6 access-list ACCESS-PORT
  remark Block all traffic DHCP server -> client
  deny udp any eq 547 any eq 546
  remark Block Router Advertisements
  deny icmp any any router-advertisement
  permit any any

interface gigabitethernet 1/0/1
  switchport
  ipv6 traffic-filter ACCESS-PORT in
```

# IPv6 ACL to Protect VTY

---

- Protecting router VTYs is very important
  - Remember: device security is as good as the least protected protocol

```
ipv6 access-list VTY
 permit ipv6 2001:db8:0:1::/64 any
!
line vty 0 4
 ipv6 access-class VTY in
```

# IPv6 Filtering

---

- IPv6 access-lists (ACL) are used to filter traffic and restrict access to the router
  - Used on router interfaces
  - Used to restrict access to the router
  - ACLs matching source/destination addresses, ports and various other IPv6 options
- IPv6 prefix-lists are used to filter routing protocol updates
  - Used on BGP peerings
  - Matching source and destination addresses

# IPv6 prefix-list example

---

- Example of using an ipv6 prefix-list to filter prefixes on a BGP session:

```
router bgp 10
  neighbor 2001:db8:1:1019::1 remote-as 20
  !
  address-family ipv6
    neighbor 2001:db8:1:1019::1 activate
    neighbor 2001:db8:1:1019::1 prefix-list ipv6-ebgp in
    neighbor 2001:db8:1:1019::1 prefix-list v6out out
    network 2001:db8::/32
  exit-address-family
  !
  ipv6 prefix-list ipv6-ebgp permit ::/0 le 128
  !
  ipv6 prefix-list v6out permit 2001:db8::/32
  !
```



# Routing Security

---

- ❑ Implement the recommendations in <https://www.routingmanifesto.org/manrs>
  1. Prevent propagation of incorrect routing information
    - ❑ Filter BGP peers, in & out!
  2. Prevent traffic with spoofed source addresses
    - ❑ BCP38 – Unicast Reverse Path Forwarding
  3. Facilitate communication between network operators
    - ❑ NOC to NOC Communication
  4. Facilitate validation of routing information
    - ❑ Route Origin Authorisation using RPKI

# Cisco IOS IPv6 NetFlow

---

- Netflow supports IPv6 as from IOS 12.4
  - Type 9 flow records
  - Following syntax in 12.4 IOS releases
- Activated by:
  - Interface subcommands:
    - `ipv6 flow ingress`
    - `ipv6 flow egress`
- Status:
  - `show ipv6 flow cache`

# IPv6 NetFlow

```
gw>show ipv6 flow cache
```

```
IP packet size distribution (520293627 total packets):
```

```
1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480
.000 .837 .130 .031 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000
```

```
512   544   576 1024 1536 2048 2560 3072 3584 4096 4608
.000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000
```

```
IP Flow Switching Cache, 475168 bytes
```

```
29 active, 4067 inactive, 11258417 added
```

```
293481382 age polls, 0 flow alloc failures
```

```
Active flows timeout in 30 minutes
```

```
Inactive flows timeout in 15 seconds
```

```
IP Sub Flow Cache, 33992 bytes
```

```
0 active, 1024 inactive, 0 added, 0 added to flow
```

```
0 alloc failures, 0 force free
```

```
1 chunk, 1 chunk added
```

SrcAddress Packets	InpIf	DstAddress	OutIf	Prot	SrcPrt	DstPrt
2001:7F8:4:1::44FC:1	Local	2001:7F8:4:1::219F:1	Gi0/0	0x06	0x00B3	0x9658 11
2001:7F8:4:1::219F:1	Gi0/0	2001:7F8:4:1::44FC:1	Local	0x06	0x9658	0x00B3 11
2001:7F8:4:1::44FC:1	Local	2001:7F8:4:1::220A:2	Gi0/0	0x06	0x00B3	0x8525 110
2001:7F8:4:1::44FC:1	Local	2001:7F8:4:1::847:1	Gi0/0	0x3A	0x0000	0x8800 14
2001:7F8:4:1::32E6:1	Gi0/0	FE80::222:55FF:FEE4:1F1B	Local	0x3A	0x0000	0x8800 256
2001:7F8:4:1::220A:2	Gi0/0	2001:7F8:4:1::44FC:1	Local	0x06	0x8525	0x00B3 82
FE80::212:F2FF:FEF2:3C61	Gi0/0	FE80::222:55FF:FEE4:1F1B	Local	0x3A	0x0000	0x8800 256
2001:7F8:4:1::1F8B:1	Gi0/0	2001:7F8:4:1::44FC:1	Local	0x06	0x00B3	0x4533 4

# Cisco IOS IPv6 Netflow (15.0+)

---

- Flexible Netflow from 12.4T and 15.0 software releases:

```
flow monitor FLOW-MONITOR-V6-IN
  exporter EXPORTER
  cache timeout active 300
  record netflow ipv6 original-input
!
flow monitor FLOW-MONITOR-V6-OUT
  exporter EXPORTER
  cache timeout active 300
  record netflow ipv6 original-output
!
interface GigabitEthernet0/0
  ipv6 flow monitor FLOW-MONITOR-V6-IN input
  ipv6 flow monitor FLOW-MONITOR-V6-OUT output
!
```

# Cisco IOS IPv6 Netflow (15.0+)

---

- Show commands are more sophisticated, for example:
  - Show the top 20 outbound IPv6 flows

```
show flow monitor FLOW-MONITOR-V6-OUT cache aggregate  
ipv6 source address ip64 destination address sort counter  
bytes top 20
```

- Show the top 20 inbound IPv6 flows

```
show flow monitor FLOW-MONITOR-V6-IN cache aggregate ipv6  
source address ipv6 destination address sort counter  
bytes top 20
```

# Securing IPv6 Connectivity



How do we secure our end-to-end connections...?

# Securing IPv6 Connectivity

---

- Over Internet
  - Client to Server:
    - IPsec or SSL VPN Client Software
  - Network to Network:
    - Tunnel technology (GRE) protected by IPsec
- Site to Site VPNs
  - Tunnel technology (GRE or MPLS) protected by IPsec

# Secure IPv6 over IPv4/6 Public Internet

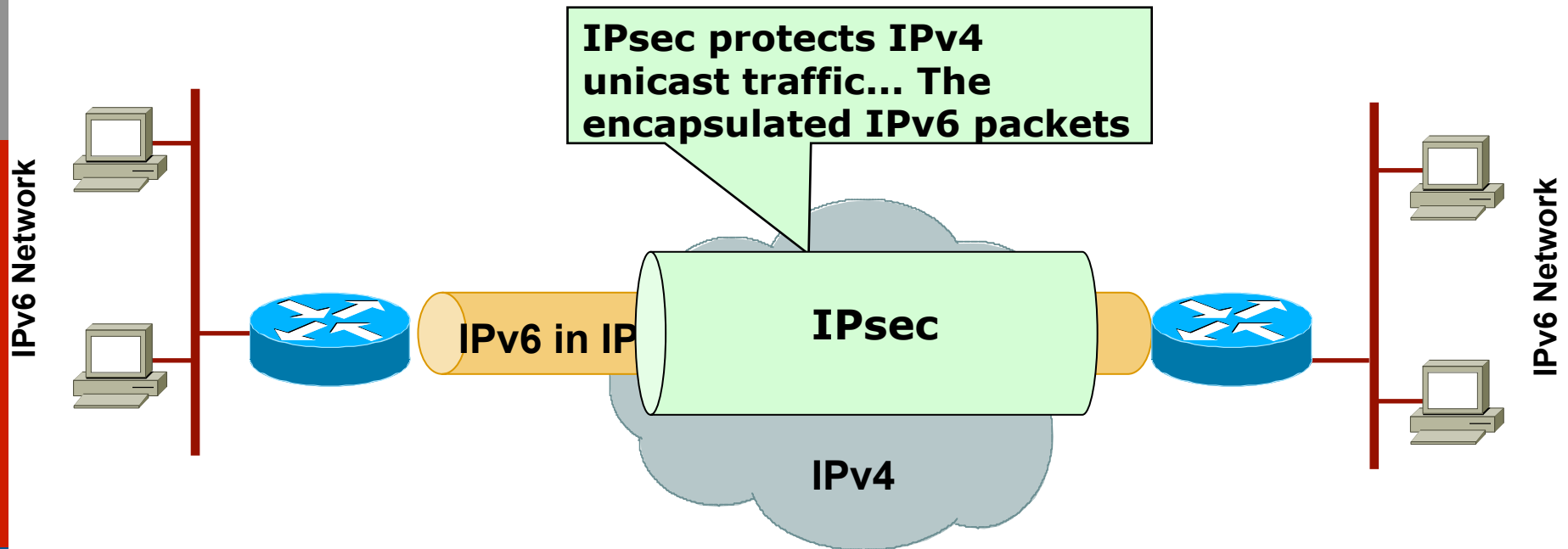
---

- ❑ No traffic sniffing
- ❑ No traffic injection
- ❑ No service theft

Public Network	Site to Site	Remote Access
IPv4	6in4/GRE Tunnels Protected by IPsec	IPsec or SSL VPN Clients
IPv6	GRE Tunnels Protected by IPsec	IPsec or SSL VPN Clients



# Secure Site to Site IPv6 Traffic over IPv4 Public Network with GRE IPsec



GRE tunnel can be used to transport both IPv4 and IPv6 in the same tunnel

# IPv6 Security Best Practices



Recommendations...

# Candidate Best Practices (1)

---

- ❑ Train your network operators and security managers on IPv6
- ❑ Train your network operators and security managers on IPv6
  
- ❑ Selectively filter ICMP (RFC 4890)
  - Might be easier to rate-limit ICMPv6 to a few Mbps
- ❑ Block Type 0 Routing Header at the edge
  - Should be automatically blocked by equipment already (but do it anyway)

# Candidate Best Practices (2)

---

- Adopt **all** the IPv4 Best Current Practices
  - **Implement BCP38 filtering**
  - Implement the Routing Security recommendations in <https://www.routingmanifesto.org/manrs>
  - If management plane is only IPv4, block IPv6 to the core devices
  - If management plane is dual stack, replicate IPv4 filters in IPv6
  - Which extension headers will be allowed through the access control device?
  - Deny IPv6 fragments destined to network equipment when possible
  - Use authentication to protect routing protocols
  - Document procedures for last-hop traceback

# Candidate Best Practices (3)

## Mainly for Enterprise Customers

---

- ❑ Implement privacy extensions carefully
- ❑ Only allow Global Unicast address sourced traffic out the border routers
  - Block ULA and other non-assigned IPv6 addresses
- ❑ Filter unneeded services at the firewall
- ❑ Maintain host and application security
- ❑ Use cryptographic protections where critical
- ❑ Implement ingress filtering of packets with IPv6 multicast source addresses
- ❑ **Avoid tunnels**
  - If you must tunnel, use static tunneling **NOT** dynamic tunneling

# Conclusion

---

- ❑ So, nothing really new in IPv6
- ❑ Lack of operational experience may hinder security for a while ⇒ **training is required**
- ❑ Security enforcement is possible
  - Control your IPv6 traffic as you do for IPv4
- ❑ Leverage IPsec to secure IPv6 when suitable

# IPv6 Security



ITU/APNIC/MICT IPv6 Security  
Workshop

23<sup>rd</sup> – 27<sup>th</sup> May 2016

Bangkok