# IPv6 Deployments

# Is Security An Afterthought (again)?

Merike Kaeo

merike@doubleshotsecurity.com

www.doubleshotsecurity.com


# Causes of Security Related Issues

- Protocol error
  - No one gets it right the first time
- Software bugs
  - Is it a bug or feature ?
- Active attack
  - Target control/management plane
  - Target data plane
  - More probable than you think !
- Configuration mistakes
  - Most common form of problem

# What Can Intruders Do?

- Eavesdrop - compromise routers, links, or DNS
- Send arbitrary messages (spoof IP headers and options)
- Replay recorded messages
- Modify messages in transit
- Write malicious code and trick people into running it
- Exploit bugs in software to ‘take over’ machines and use them as a base for future attacks

# Security Services

- User Authentication / Authorization
- Device Authentication / Authorization
- Access Control (Packet Filtering)
- Data Integrity
- Data Confidentiality
- Auditing / Logging
- DoS Mitigation

# What Is The Same / What Is Different

- Same for IPv4 and IPv6
  - Security Properties
  - Security Services
- Different for IPv6 Architectures
  - Protocol Operation
  - More Automation
  - Scalable Mobile Hosts
  - Potential Application Integration

# What Needs To Be Considered

- Where is automation advantageous versus a security risk?
- How will IPv4 content be accessible?
  - Is NAT a security feature or a simple way of getting access to the global Internet (without paying for it)?
  - Where is an address translation capability required?
- Where are network-based security mitigation techniques reliably advantageous versus a hindrance?
- What technologies need to be made easier to deploy to be operationally viable?
- What security services are being used to adhere to security policy requirements but are instantiations of IPv4 architecture limitations?

# IPv6 Automation

- Protocol Capabilities
  - Neighbor Discovery allows nodes to easily find one another
  - Router Advertisements enable nodes to automatically create their own globally reachable IPv6 address
- Security Issues
  - Redirect attacks
  - Denial-of-Service attacks
  - Neighbor solicitation spoofing
  - Neighbor advertisement spoofing
  - Neighbor Unreachability Detection failure
  - Duplicate Address Detection DoS attack

# Architecture Considerations

- Addressing / Naming
  - What subnet boundaries make sense
    - your own network infrastructure
    - filtering considerations
  - Endpoint Identifier management
    - address automation vs obscurity vs auditability
  - DNS and DHCPv6 Considerations
- Native Routing vs Tunnels
- Management
- Security (Is This A Last Consideration In Practice?)

# Required Host IPv6 Addresses

- Each host must assign the following addresses to identify itself:
  - Its link-local address for each interface
  - Any assigned unicast addresses
  - The loopback address
  - The all-nodes multicast address
  - Solicited-node multicast address for each assigned unicast or anycast address
  - Multicast addresses for all other group memberships

# Stateless Address Autoconfiguration (SLAAC)

- RFC2462
- For autoconfiguration of IPv6 there are two options
  - Stateful (DHCPv6)
  - Stateless (via RA)
- For SLAAC this is done by combining address prefix advertised in the RA with the Interface ID
  - EUI-64 or RFC3041 (privacy addresses)
- Thought to help renumbering of a network
- Problem
  - How do I find a DNS server?
  - How do I send update to the DNS server?

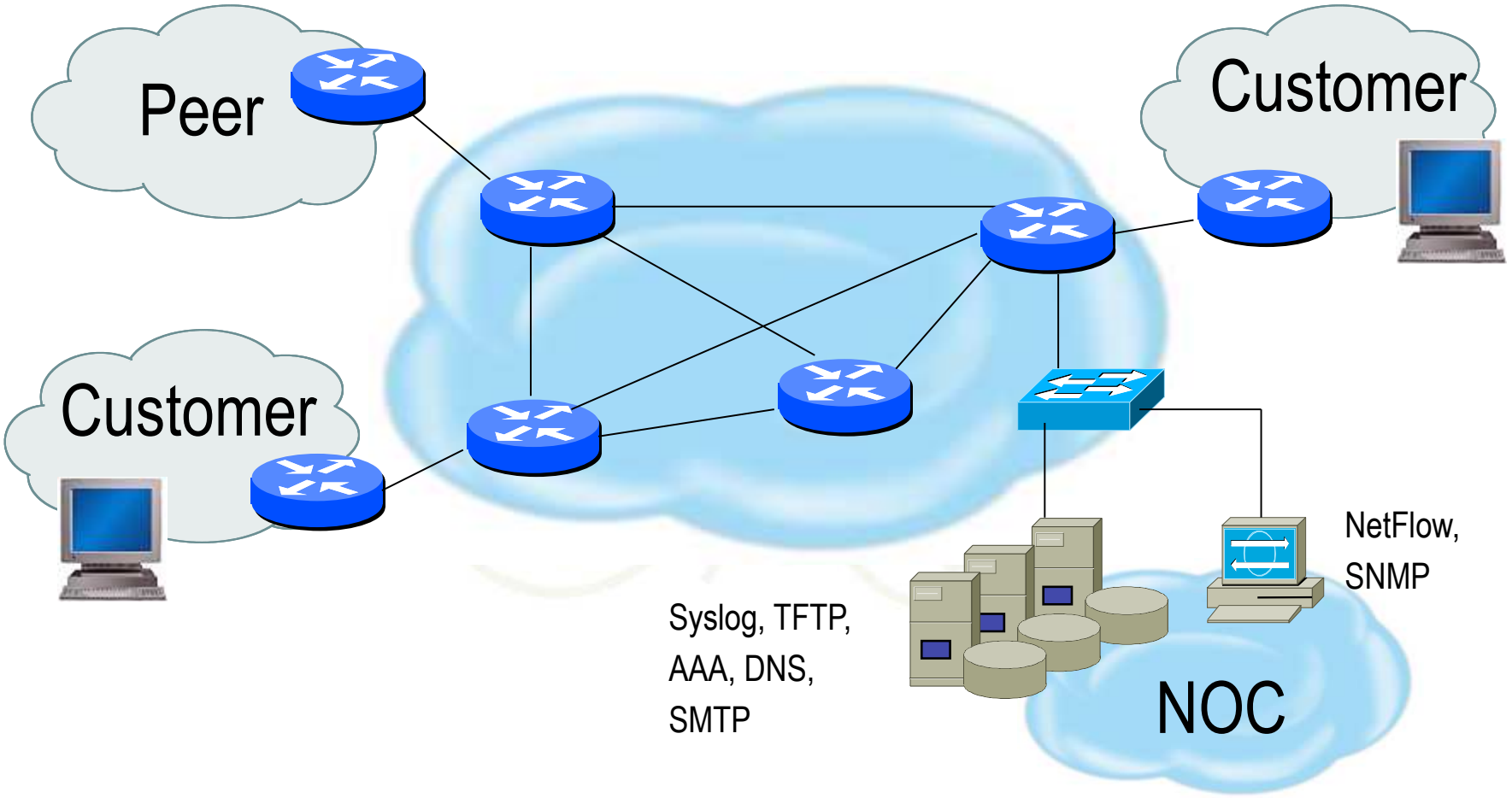
# Tunneling Considerations

- Manually configured tunnels are not scalable
- Automated tunnels require more diligence to provide effective security services
- Deployments of any transition technologies all require layered security models
  - Perform ingress firewall sanity checks
  - Log and audit tunneled traffic
  - Provide authentication where possible
  - Use IPsec where appropriate

# IPv6 Security Enhancements

- Fragmentation
  - Prohibited by intermediary devices
  - Overlapping fragments are not allowed
  - Devices must drop reassembled packets that are less than 1280 bytes
- Broadcasts
  - Removes concept of dedicated broadcasts
  - Specific language to avoid ICMPv6 broadcast amplification attacks
- IPsec
  - Defined into the base protocol spec

# Infrastructure Security



# Fundamental Issues

- What is meant by *Securing The Network* ?
- Design security into IPv6 networks that do not blindly mimic the current IPv4 architectures
  - Don't break working v4 infrastructure
  - Don't re-architect current problems and place limitations on IPv6 capabilities
- Requires some thought to policy
  - Where are you vulnerable today ?
  - What new application capabilities are possible with IPv6?
  - New risk assessment will help (re)define appropriate security policy
- Security policy will dictate which security measures to implement


# IPv6 Security Theory vs Reality

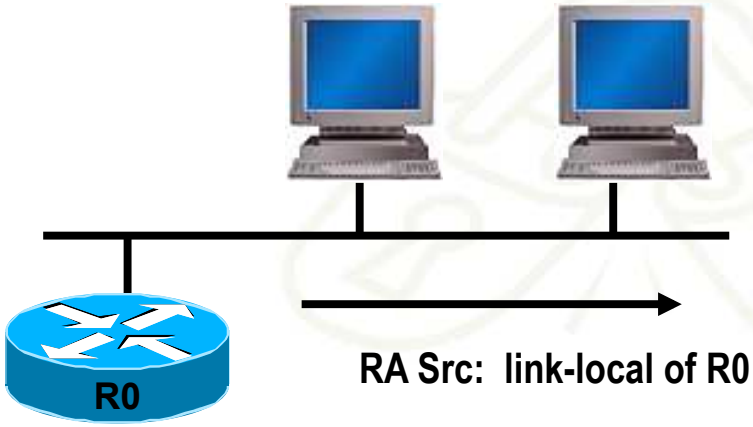
- IPv6 has security built-in
  - Mostly based on mandate to implement IPsec
  - IPsec use was never fully defined in IPsec specs
    - Early implementations made it up
    - Configuration is still difficult and often operationally not optimal
    - IPv6 conformance testing doesn't necessarily require it
- IPv6 needs IPv4 security feature parity
  - Yes and No ☺

# IPv6 Address Resolution

- Neighbor Discovery (ND) replaces the functionality of the Address Resolution Protocol in IPv4
- All ND messages are encapsulated using ICMP transport and are identified by ICMP types
- All ND messages have the hop limit field set to 255
- ND uses ICMPv6 to perform the following functions:
  - router discovery
  - prefix discovery
  - auto-configuration of addresses and other parameters
  - address renumbering
  - duplicate address detection (DAD)
  - neighbor unreachability detection (NUD)
  - link-layer address resolution
  - first-hop redirect

# IPv6 Router Advertisement

- Sent periodically or in response to a Router Solicitation message
- Periodic RA's are sent to the all-nodes multicast address "ff02::1"
- RA messages contain information that inform the hosts about link information needed for auto-configuration




**RA Src: link-local of R0 interface**

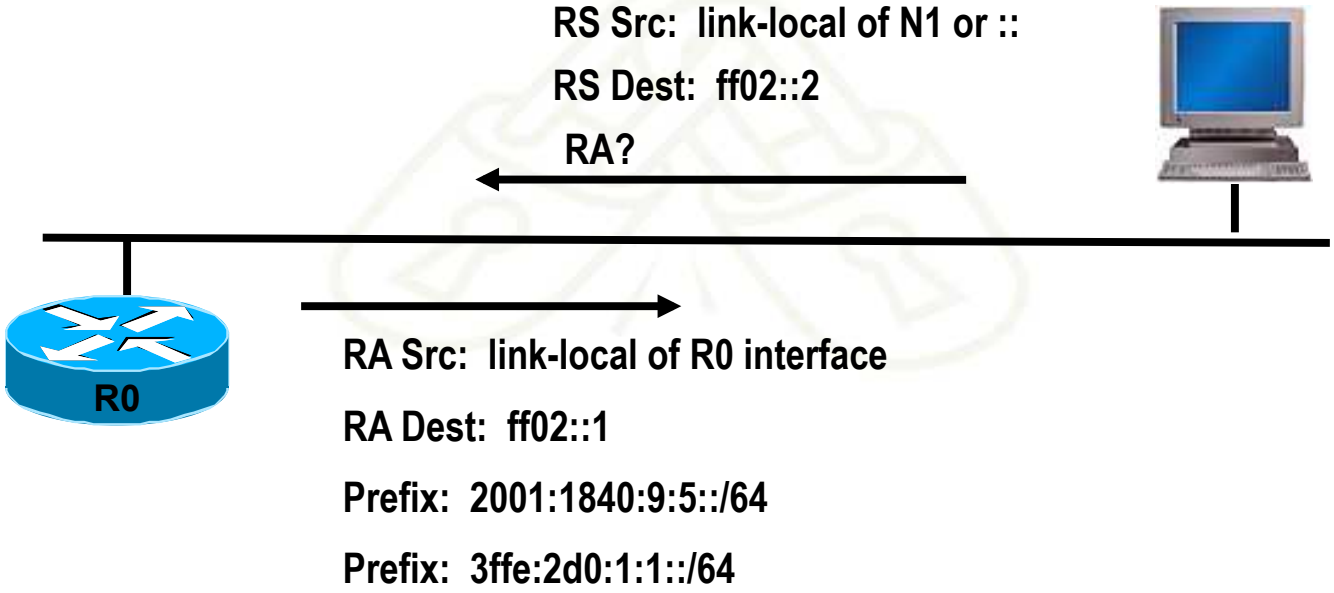
**RA Dest: ff02::1**

**Prefix: 2001:1840:9:5::/64**

**Prefix: 3ffe:2d0:1:1::/64**

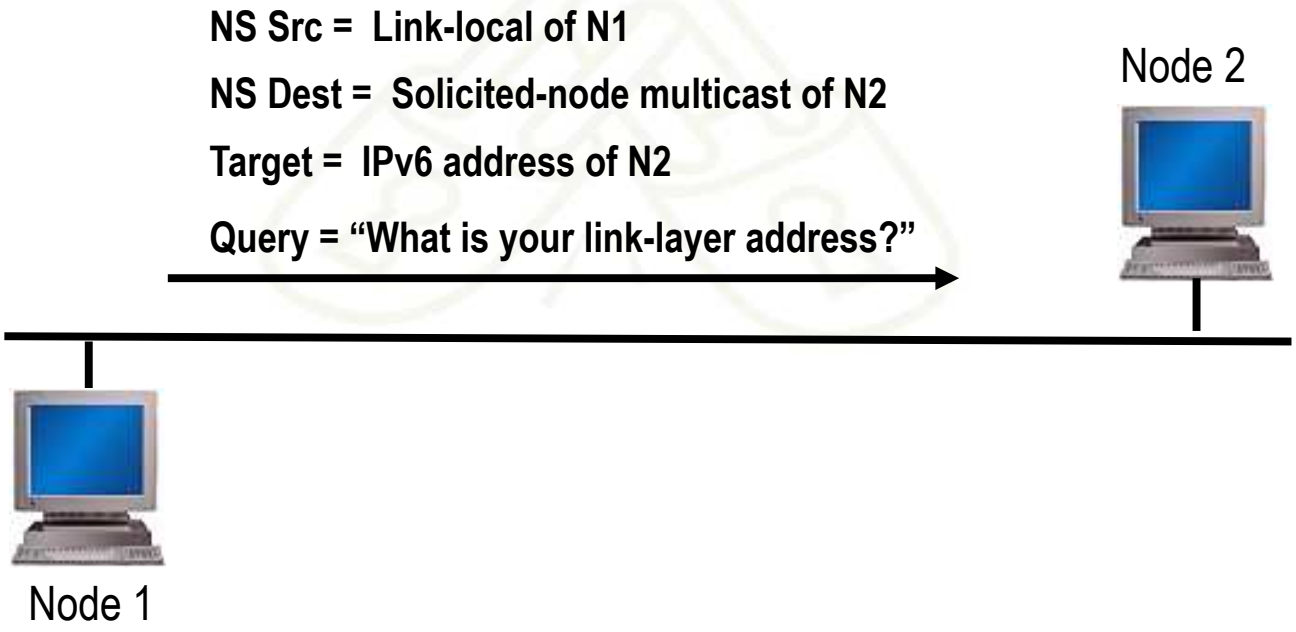
# IPv6 Router Solicitation

- Sent at host start-up or to solicit a Router Advertisement immediately
- RS messages are usually sent to the all-routers multicast address “ff02::2”
- RS source address could be the link-local address of the sending node, or the unspecified “::” address



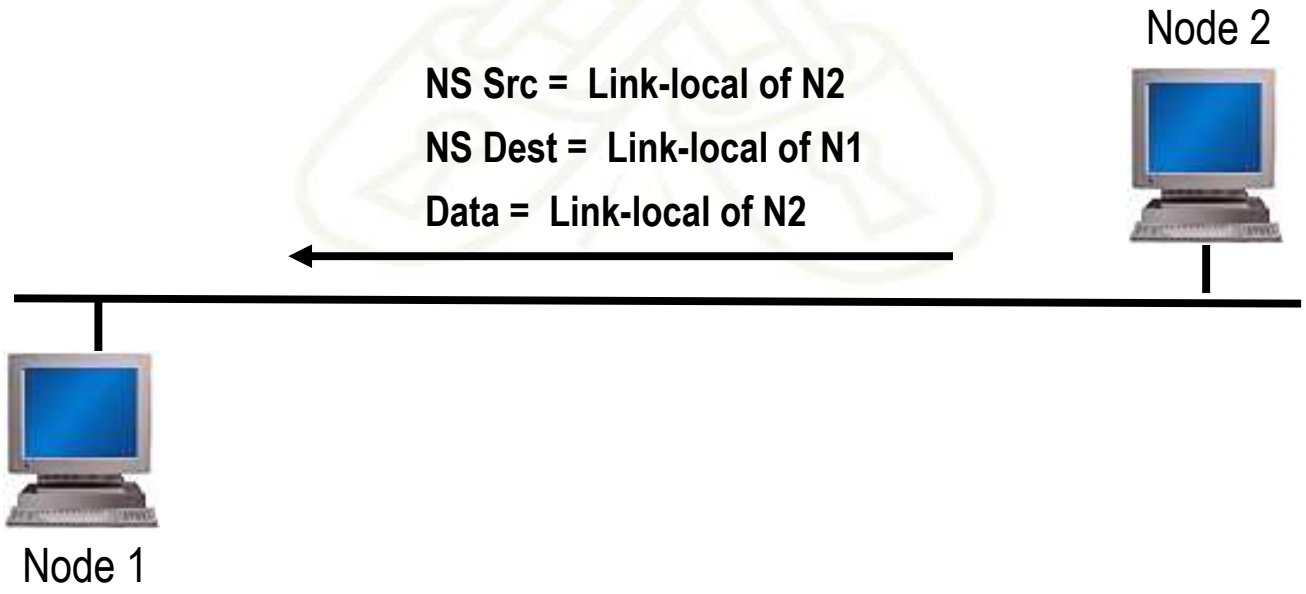
# IPv6 Neighbor Solicitation

- Used by nodes for link-layer to IP-layer address resolution
- For link-layer address resolution, the solicited-node multicast address is used as the destination of the request (vs. broadcast in IPv4 ARP)
- Also used in the Duplicate Address Detection (DAD) and Neighbor Unreachability Detection (NUD) processes

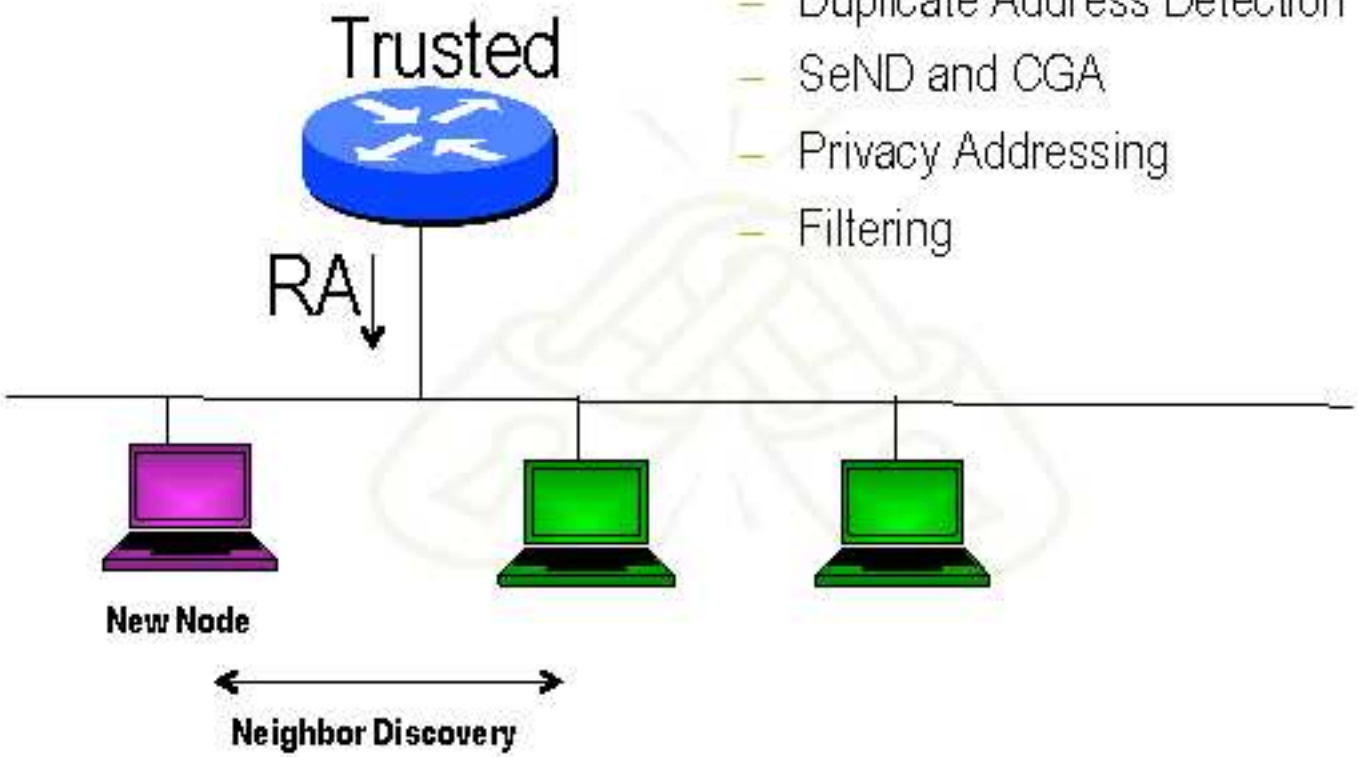


# IPv6 Neighbor Advertisement

- Sent in response to an NS or unsolicited to propagate new information
- Neighbor Advertisements contain:
  - Router flag: to indicate whether this neighbor is a router
  - Solicited flag: to indicate whether this NA is in response to a NS
  - Override flag: to indicate whether this information should override an existing neighbor cache entry
- NA's in response to an address resolution request are unicast to the solicitor

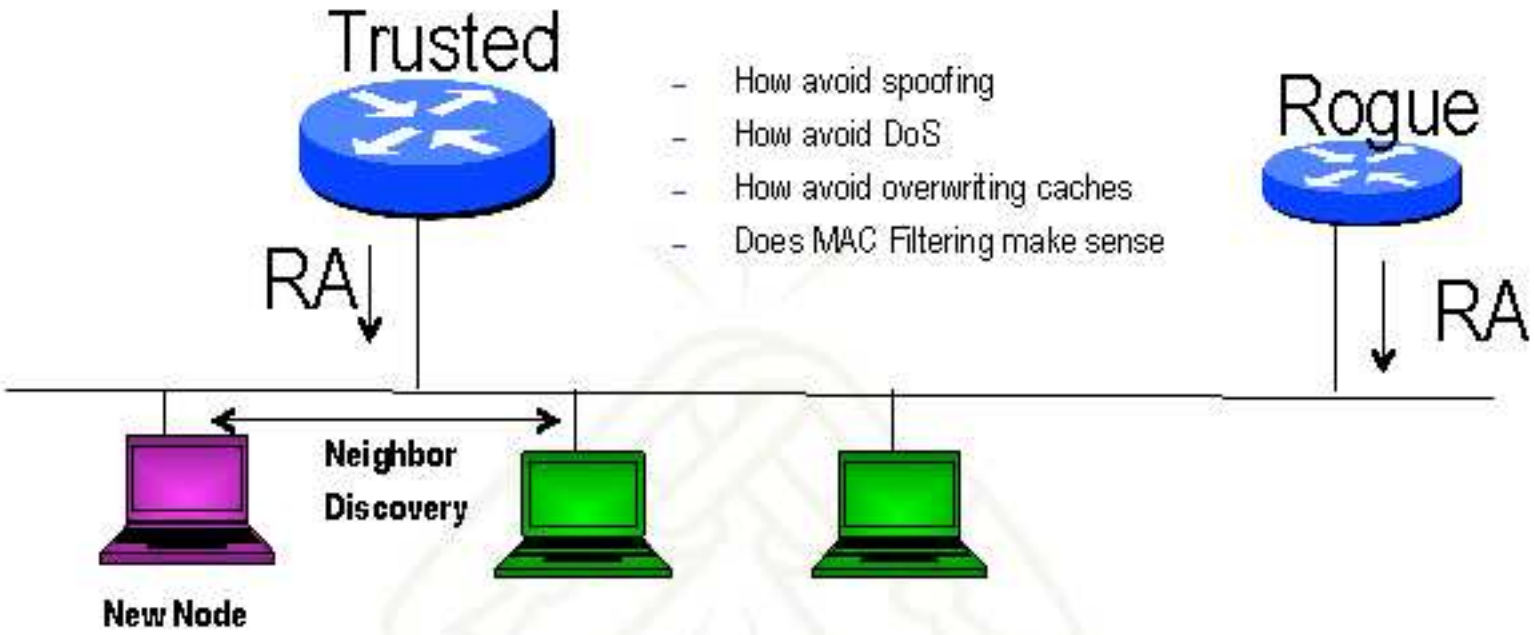


# Node Initialization Security (Theory)




- Duplicate Address Detection
- SeND and CGA
- Privacy Addressing
- Filtering

# Node Initialization Security (Practice)




- How avoid spoofing
- How avoid DoS
- How avoid overwriting caches
- Does MAC Filtering make sense

- Host behaviors vary and need to be understood
- SeND and CGA not widely used (yet?)
- Layer 2 mitigation techniques wip for vendors
- http://www.kb.cert.org/vuls/id/472363

# SeND (Secure Neighbor Discovery)

- Hosts configured with trust anchors
- Trust anchor
  - Entity trusted to authorize routers to act as routers
  - Public key and associated parameters
  - Certification path solicitation and advertisement messages used to discover path to a trust anchor out-of-band
- Cryptographically generated addresses are used to make sure sender is 'owner' of claimed address (optional)
  - Interface identifier is generated by computing a cryptographic one-way hash from public key and associated parameters
- Public/private key pair is generated by all nodes before they can claim an address

# SeND (Secure Neighbor Discovery)

- ND RSA public key signatures used to protect all messages
  - protects integrity of message
  - authenticates identity of sender
- Authority of public key is established by
  - authorization delegation process by using certificates
  - address ownership proof mechanism by using CGAs
- Replay protection through use of timestamp (multicast) and nonce (communicating pair)


# SeND Capabilities

- SeND protects against:
  - Spoofed Messages To Create False Entries In Neighbor Cache
  - Neighbor Unreachability Detection Failure
  - Duplicate Address Detection DoS Attack
  - Router Solicitation and Advertisement Attacks
  - Replay Attacks
  - Neighbor Discovery DoS Attacks
- SeND does NOT:
  - Protect statically configured addresses
  - Protect addresses configured using fixed identifiers (I.e.EUI-64)
  - Provide confidentiality
  - Compensate for unsecured link-layer
    - No guarantee that payload packets came from node that used SEND


# Node Global Addressing Security (Theory)

- Static addressing can be used
- Stateful Autoconfiguration
  - Requires use of a server to give hosts information
- Stateless Autoconfiguration
  - Requires no manual configuration of hosts
  - Minimal (if any) configuration on routers
- Privacy Addresses (rfc4941)
- Router Advertisements vs DHCPv6

# Node Global Addressing Security (Practice)

- Statically defined addresses used for critical devices
- Privacy addresses are used by default by Vista
  - How do you correlate IPv6 address to log info?
- Router Advertisement
  - Relying on unauthenticated broadcast packet to determine where host should send traffic to
- DHCPv6
  - Can send requests to local LAN before get an RA message telling you to do so. This requires manual configuration on host


# Better RA/DHCPv6 Filtering Needed

- Networks with visitors have shown a serious problem with rogue RA and DHCP servers
  - Networks with visitors that use either RA or DHCPv6 for address assignment will have the exact same problem if someone comes along with a rogue server
- Features needed to limit where RA messages and DHCPv6 messages can be sent from
  - Allow RA messages only from routers, and DHCPv6 responses only from DHCPv6 servers
- Some Ethernet equipment has the ability to filter on Ethernet source/destination
  - Only allow messages to the all routers multicast address to go to the switch interfaces that have routers on them
  - Only allow messages to the all DHCPv6 servers multicast address to go to the switch interfaces that have DHCPv6 servers or relays on them


# Securing The Device (same in IPv4 and IPv6)

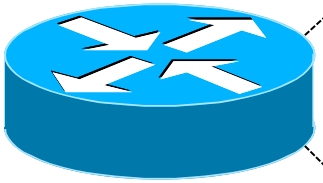
# Device Access

- Console Port
  - Access via cable connected to the serial port
  - Only access to password recovery functions
- Auxiliary Port
  - Generally used for out of band (OOB) access
  - Also used for connecting to other console ports
- Virtual TTY (VTY)
  - Default access is via ‘telnet’
- HTTP
- TFTP
- SNMP


# Access Control Best Practices

- Set passwords to something not easily guessed
- Use single-user passwords (avoid group passwords)
- Encrypt the passwords in the configuration files
- Use different passwords for different privilege levels
- Use different passwords for different modes of access

# Secure Access with Passwords and Logout Timers



```
line console 0
    login
    password console-pw
    exec-timeout 1 30
line vty 0 4
    login
    password vty-pw
    exec-timeout 5 00

enable secret enable-secret
username merike secret merike-secret
```


# Never Leave Passwords in Clear-Text

- ***service password-encryption*** command
- ***password*** command
  - Will encrypt all passwords on the Cisco IOS with Cisco-defined encryption type “7”
  - Use “*command* password 7 <password>” for cut/paste operations
  - Cisco proprietary encryption method
- ***secret*** command
  - Uses MD5 to produce a one-way hash
  - Cannot be decrypted
  - Use “*command* secret 5 <password>” to cut/paste another “enable secret” password

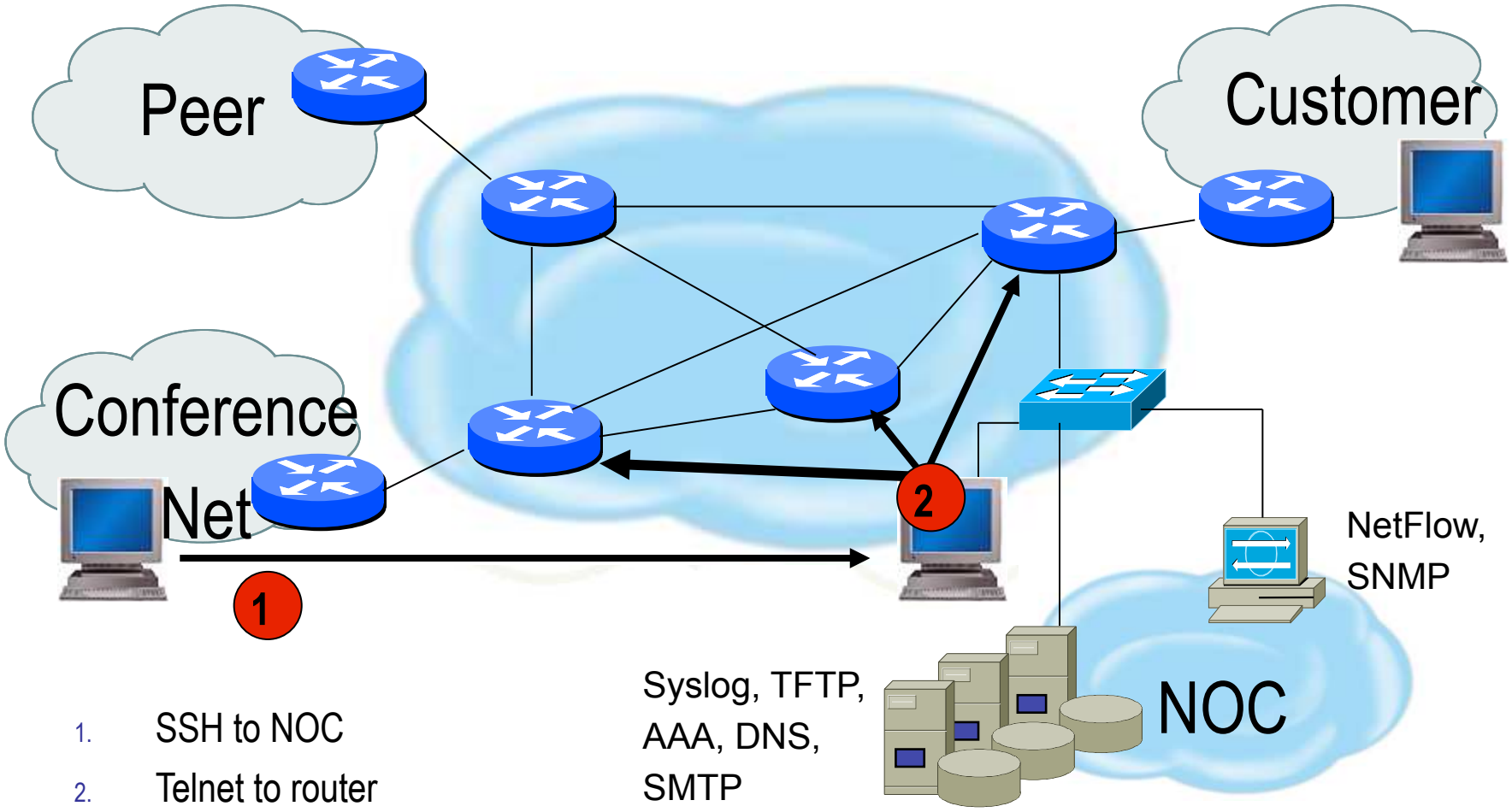
# Management Plane Filters

- Authenticate Access
- Define Explicit Access To/From Management Stations
  - SNMP
  - Syslog
  - TFTP
  - NTP
  - AAA Protocols
  - DNS
  - SSH, Telnet, etc.

# Telnet is Insecure

- Avoid using Telnet if possible
- Telnet sends username and password information across the wire in plain text format.
- Do not use telnet to gain access to any of your boxes
- Use jumphosts for legacy equipment

# Telnet using SSH 'Jumphost'




1. SSH to NOC
2. Telnet to router


# Secure Shell (SSH)

- Username/password information is encrypted
- Flexible authentication methods
  - One-time password
  - Kerberos
  - Public key
- Allows Secure Tunneling
  - TCP port forwarding
  - Forward remote ports to local ones
- Uses TCP port 22

# SSH Support

- Two flavors of ssh, ssh1 and ssh2
- Use ssh2 if possible
- In general the client connecting to your ssh server will either "speak" ssh1 or ssh2
- OpenSSH for UNIX
  - www.openssh.org
  - Supports both ssh1 and ssh2
- Putty client for Windows
  - www.chiark.greenend.org.uk/~sgtatham/putty/


# Added Controls For SSH Access

Configure IPv6 vty-input access-list

ipv6 access-list vty-filter

permit host <ipv6 address> host <ipv6 address>

Apply vty-input access-list to vty 0 4

line vty 0 4

ipv6 access-class vty-filter in


# Secure SNMP Access

- SNMP is primary source of intelligence on a target network!
- Block SNMP from the outside

  *access-list 101 deny udp any any eq snmp*
- If the router has SNMP, protect it!

  *snmp-server community fO0bAr RO 8*

  *access-list 8 permit 127.1.3.5*
- Explicitly direct SNMP traffic to an authorized management station.

  *snmp-server host fO0bAr 127.1.3.5*

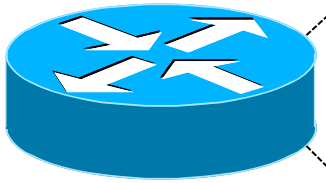
# SNMP Best Practices

- SNMP over IPv6 transport is not widely available but until you have devices that speak IPv6 only it's not an issue
- For now, SNMP will use IPv4 transport
  - Do not enable read/write access unless really necessary
  - Choose community strings that are difficult to guess
  - Limit SNMP access to specific IP addresses
  - Limit SNMP output with views

# Secure Logging Infrastructure

- Log enough information to be useful but not overwhelming.
- Create backup plan for keeping track of logging information should the syslog server be unavailable
- Remove private information from logs
- How accurate are your timestamps?

# Banner – What Is Wrong ?

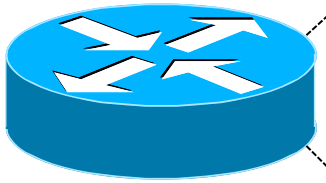


```
banner login ^C

    You should not be on this device.

    Please Get Off My Router!!
^C
```

# More Appropriate Banner



!!!! WARNING !!!!

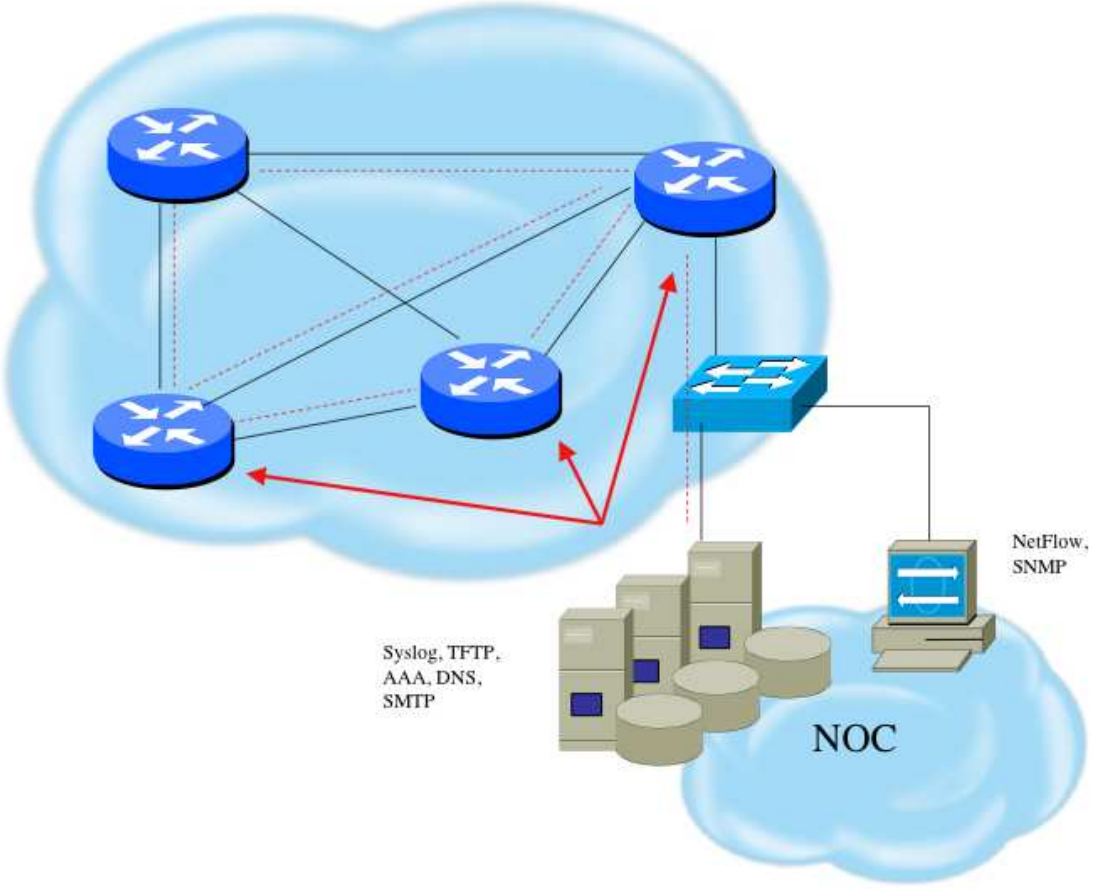
You have accessed a restricted device.

All access is being logged and any unauthorized access will be prosecuted to the full extent of the law.


# Turn Off Unused Services

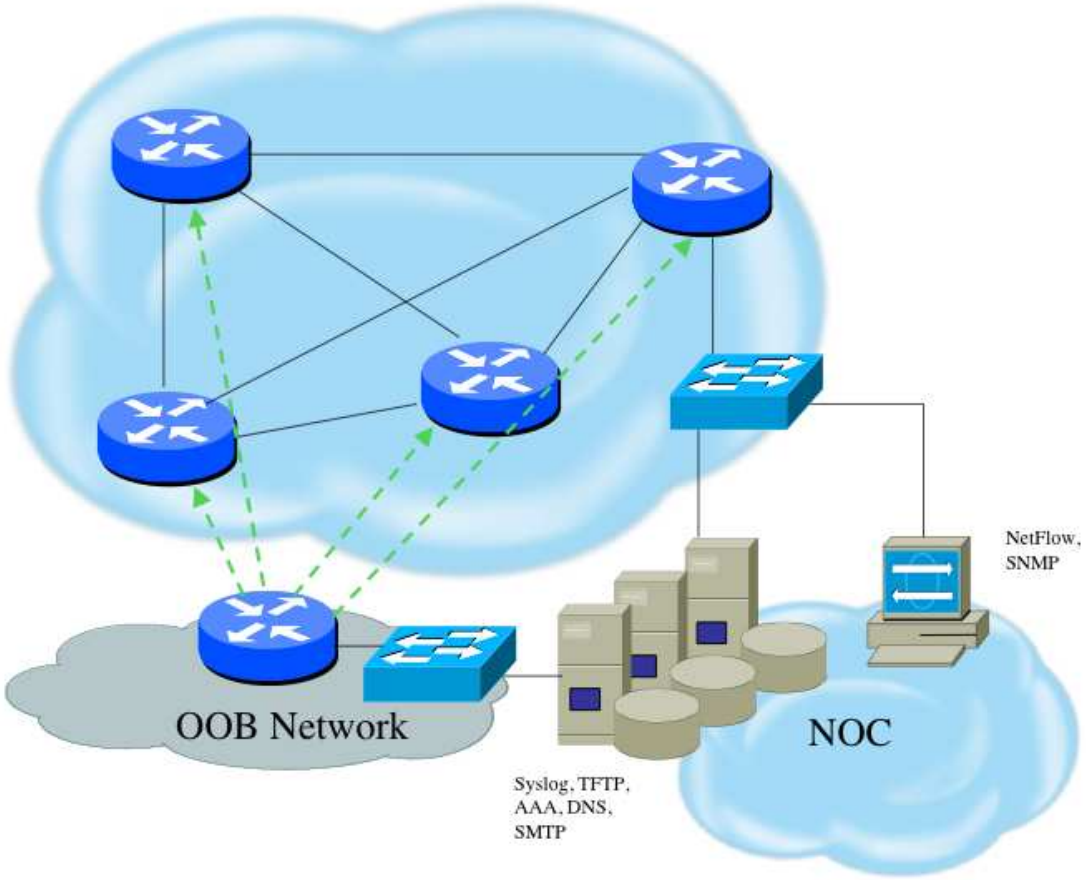
- Global Services
  - no service finger (before 12.0)
  - no ip finger
  - no service pad
  - no service udp-small-servers
  - no service tcp-small-servers
  - no ip bootp server
  - no cdp run
- Interface Services
  - no ip redirects
  - no ip directed-broadcast
  - no ip proxy arp
  - no cdp enable

# Device In-Band Management




- Management traffic uses same path as transit data
- Usually an issue of operational cost

# Device OOB Management



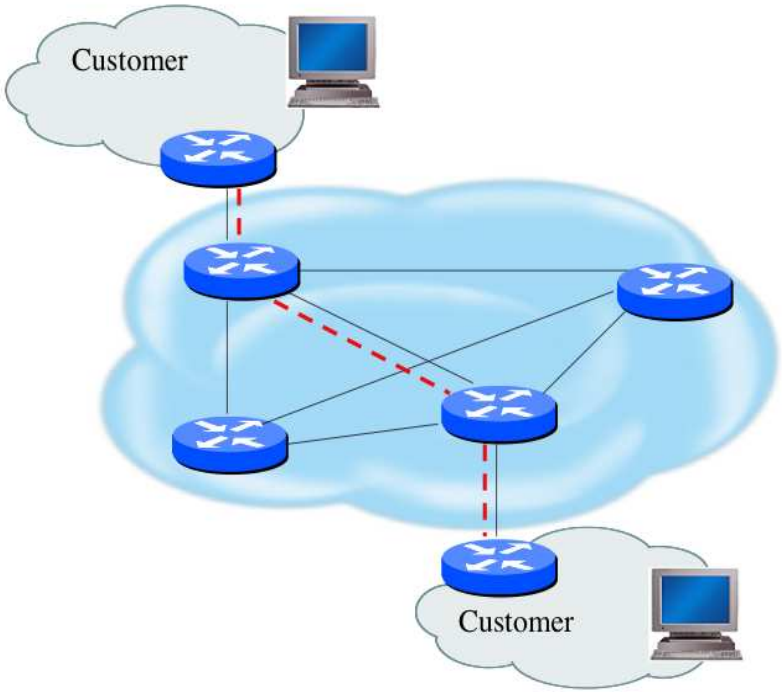

- Terminal servers are used at each location for OOB management
- Dial-back encrypted modems are used as backup

# Device Management Same in v4 and v6

- SSH primarily used; Telnet only from jumphosts
- HTTP access explicitly disabled
- All access authenticated
  - Varying password mechanisms
  - AAA usually used
    - Different servers for in-band vs OOB
    - Different servers for device authentication vs other
    - Static username pw or one-time pw
  - Single local database entry for backup
- Each individual has specific authorization
- Strict access control via filtering
- Access is audited with triggered pager/email notifications
- SNMP is read-only
  - Restricted to specific hosts
  - View restricted if capability exists
  - Community strings updated every 30-90 days


# Securing The Data

# Securing The Data Path




- Filtering and rate limiting are primary mitigation techniques
- BCP-38 guidelines for ingress filtering
- Null-route and black-hole any detected malicious traffic
- Netflow is primary method used for tracking traffic flows
- Unicast Reverse Path Forwarding is not consistently implemented
- Logging of Exceptions


# Data Plane (Packet) Filters

- Most common problems
  - Poorly-constructed filters
  - Ordering matters in some devices
- Scaling and maintainability issues with filters are commonplace
- Make your filters as modular and simple as possible
- Take into consideration alternate routes
  - Backdoor paths due to network failures


# Filtering Deployment Considerations

- How does the filter load into the router?
- Does it interrupt packet flow?
- How many filters can be supported in hardware?
- How many filters can be supported in software?
- How does filter depth impact performance?
- How do multiple concurrent features affect performance?
- Do I need a standalone firewall?

# Filtering Recommendations

- Log filter port messages properly
- Allow only internal addresses to enter the router from the internal interface
- Block packets from outside (untrusted) that are obviously fake or commonly used for attacks
- Block packets that claim to have a source address of any internal (trusted) network.

# RFC2827 (BCP38) – Ingress Filtering

If an ISP is aggregating routing announcements for multiple downstream networks, strict traffic filtering should be used to prohibit traffic which claims to have originated from outside of these aggregated announcements.

The ONLY valid source IP address for packets originating from a customer network is the one assigned by the ISP (whether statically or dynamically assigned).

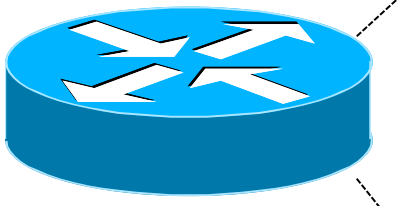
An edge router could check every packet on ingress to ensure the user is not spoofing the source address on the packets which he is originating.


# IPv6 Filtering Considerations

- IPv6 addressing architecture will simplify or complicate filters….carefully think about it.
- Routing filters are usually more optimal than packet filters but have less granularity
  - Routing filters affect the routes that are accepted and sent between routers and therefore forward or drop traffic based on reachability information
  - Packet filters are used to allow or deny data packets from being processed or forwarded by a device based on the IP header information.
- Best policy is to deploy filtering mechanisms that will drop any unwanted traffic as close to source as possible

# Packet and/or Route Filtering in IPv6

- In theory, certain addresses should not be seen on the global Internet
- In practice, they are and filters aren't being deployed (even when capability available)



```
ipv6 access-list extended DSL-ipv6-Outbound
 permit ipv6 2001:DB8:AA65::/48  any
 deny   ipv6 any any log

interface atm 0/0
  ipv6 traffic-filter DSL-ipv6_Outbound out
```


# General Firewall BCP (same for IPv6 and IPv4 networks)

- Explicitly deny all traffic and only allow what you need
  - This doesn't really work in practice, especially for ISPs
- The default policy should be that if the firewall doesn't know what to do with the packet, deny/drop it
- Don't rely only on your firewall for all protection of your network
- Implement multiple layers of network protection
- Make sure all of the network traffic passes through the firewall
- Log all firewall exceptions (if possible)

# Ingress IPv6 Packet Filters To Consider

- Accept all ICMPv6 packets for Neighbor Discovery and Path MTU Discovery that is a function necessary for the communication with IPv6
  - Allow link-local (fe80::/10) as source and destination
  - Allow multicast (ff02::/16) as destination
- Reject the packets which contain relevant special-use prefix in the ***source*** address field
  - ::1/128 : loop back address
  - ::/128 : unspecified address
  - ::/96 : IETF reserved address;IPv4-compatible IPv6 address
  - ::ffff:0:0/96 : IPv4-mapped IPv6 address
  - ::/8 : reserved
  - fc00::/7 : unique-local address
  - ff00::/8 : multicast address
  - 2001:db8::/3 : documentation addresses


# Ingress IPv6 Packet Filters To Consider (2)

- Reject the packets which contain relevant special-use prefix in the ***destination*** address field
  - ::1/128 : loop back address
  - ::/128 : unspecified address
  - ::/96 : IETF reserved address;IPv4-compatible IPv6 address
  - ::ffff:0:0/96 : IPv4-mapped IPv6 address
  - ::/8 : reserved
  - fc00::/7 : unique-local [fc00::/16] and site-local [fc00::/10] address
  - 2001:db8::/32 : documentation address
- Reject the packets which have your own prefix in the source address field
- Reject packets that use the routing header.
- Care must be taken not to reject ICMPv6 packets whose source address used with Duplicate Address Detection is the unspecified address (::/128). If all of ICMPv6 is accepted, then there is no problem although ordering of the filters needs to be carefully thought through.


# Egress IPv6 Packet Filters To Consider

- Permit sending all ICMPv6 packets for Neighbor Discovery and Path MTU Discovery that is a function necessary for the communication with IPv6
- Deny sending the packets which contain special-use prefix in the source address field
  - ::1/128 : loop back address
  - ::/128 : unspecified address
  - ::/96 : IETF reserved address;IPv4-compatible IPv6 address
  - ::ffff:0:0/96 : IPv4-mapped IPv6 address
  - ::/8 : reserved
  - fc00::/7 : unique-local address
  - ff00::/8 : multicast address
  - 2001:db8::/32 : documentation address
- Deny sending packets that use the routing header [unless using mobility features]
- Deny sending packets with destination address in the 6to4 reserved address range (2202::/16) if not supporting 6to4 services (i.e. relays) and not providing transit services
- Deny sending packets with destination address in the Teredo address range (2001::/32) if not running a Teredo relay or offering a Teredo transit service
- Multicast address should only be in source address field.

# RFC4980 – ICMPv6 Filtering

- In general, Internet Service Providers should not filter ICMPv6 messages transiting their sites so that all the necessary communication elements are available to their customers to decide and filter according to their policy.
- For firewall/bridges, the physical links on either side of the firewall/bridge are treated as a single logical link for the purposes of IP. Hence, the link local messages used for discovery functions on the link must be allowed to transit the transparent bridge.

# Allow Following ICMPv6 Through Firewall

- ICMPv6 type 1 code 0: no route to destination
- ICMPv6 type 2: packet too big (required for PMTUD)
- ICMPv6 type 3: time exceeded
- ICMPv6 type 4: parameter problem (informational when IPv6 node has problem identifying a field in the IPv6 header or in an extension header)
- ICMPv6 type 128: echo request
- ICMPv6 type 129: echo reply

# Allow Following ICMPv6 To/From A Firewall

- ICMPv6 type 2: packet too big – firewall device is not allowed to fragment IPv6 packets going through it and must be able to generate this message for correct PMTUD behavior
- ICMPv6 type 4: parameter problem
- ICMPv6 type 130-132: multicast listener messages – in IPv6 a routing device must accept these messages to participate in multicast routing
- ICMPv6 type 133-134: router solicitation and advertisement – needed for IPv6 autoconfiguration
- ICMPv6 type 135-136: neighbor solicitation and advertisement – used for duplicate address detection and layer2-to-IPv6 address resolution

# Need Better IPv6 Extension Header Filtering

- Carry the additional options and padding features that are part of the base IPv4 header
- Extension headers are optional and placed after the base header
- There can be zero, one, or more Extension Headers between the IPv6 header and the upper-layer protocol header
- Ordering is important

**Currently Defined IPv6 Extension Headers:**

- Hop-by-Hop Options (0)
- Routing Header (43)
- Fragment Header (44)
- ESP Header (50)
- Authentication Header (51)
- Destination Options (60)

**Other Extension Header Values:**

- TCP upper-layer (6)
- UDP upper-layer (17)
- ICMPv6 (58)
- No Next Header Present (59)

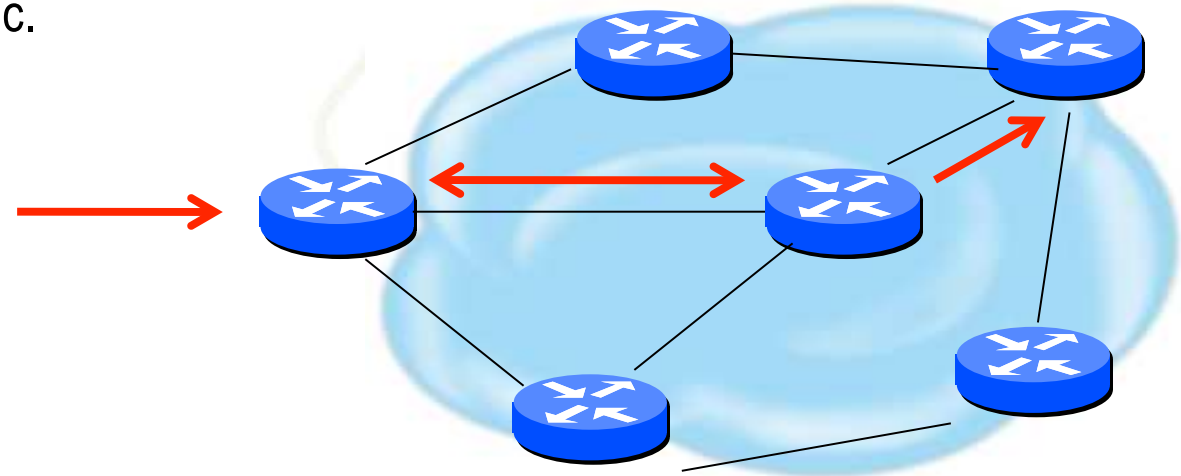
# Routing Header: RFC 2460 Text

- The routing header is used by an IPv6 source to list one or more intermediate ***nodes*** to be “visited” on the way to packet’s destination.
- Each extension header should occur at most once, except for the destination options header which should occur at most twice.
- IPv6 nodes must accept and attempt to process extension headers ***in any order*** and ***occurring any number of times*** in the same packet.

# Routing Header Issue

A single RH of Type 0 may contain multiple intermediate node addresses, and the same address may be included more than once in the same RH0.

If the routing header contains a repetition of a pair of addresses of the form A B A B A B … If this A B pair were repeated 3 times then a single packet directed at A would traverse the path A B 3 times, and B A twice. If such packets were generated at a total rate of 1 Mbps then the path between A and B would experience a total of 5Mbps of traffic.



# Routing Header Processing

- Disabling processing still allows all other hosts to be used for attack
- Dropping is required for ISP's
- RFC 5095 – Deprecation of RH0
- Until rfc5095 implemented:
  - Use ingress filtering for RH0 traffic
  - RH Type 2 is required for mobility so have to ensure that only RH0 traffic is blocked


# Cisco and RH0 Filtering

- To disable processing of all types routing headers on 12.2(15)T and up one can use:

```
no ipv6 source-route
```

*Note that this will still forward these packets on to other hosts which can be vulnerable. This statement also affects perfectly valid Routing Headers of Type 2 which are used by Mobile IPv6.*

- If possible upgrade to 12.4(2)T or higher and block only the Type 0 Routing Header (note interface specific config):

```
Router(config)#ipv6 access-list deny-sourcerouted
Router(config-ipv6-acl)#deny ipv6 any any routing-type 0
Router(config-ipv6-acl)#permit ipv6 any any
Router(config)#interface Ethernet0
Router(config-if)#ipv6 source-route
Router(config-if)#ipv6 traffic-filter deny-sourcerouted in
```


# Cisco IPv6 NetFlow

- Netflow IPv6 support from 12.4 IOS releases
- Uses Netflow v9
- Activate per interface

  ipv6 flow ingress

  ipv6 flow egress
- Show status

  show ipv6 flow cache

# IPv6 Filtering References

- RFC 4890 ‘Recommendations for Filtering ICMPv6 Messages in Firewalls’
- RFC 5156 ‘Special-Use IPv6 Addresses’
- http://www.space.net/~gert/RIPE/ipv6-filters.html
- http://www.cymru.com/Bogons/v6top.html
- NSA Router Security Configuration Guide Supplement – Security for IPv6 Routers

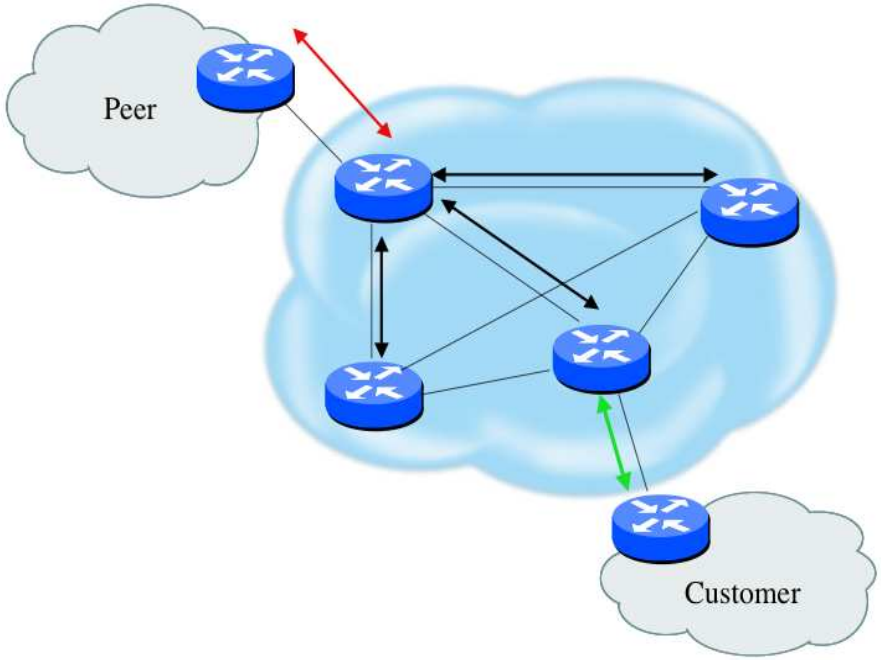
*Many filtering recommendations are not uniform and that while similarities exist, a definitive list of what to deny and what to permit does not exist. Any environment will need to determine what is most suitable for them by using these references as guidelines.*


# Securing The Routing Infrastructure


# Router Security Considerations

- Segment areas for route redistribution and ensure limited access to routers in critical backbone areas
- Design networks so outages don't affect entire network but only portions of it
- Control router access….watch against internal attacks on these systems. Use different passwords for router enable and monitoring system root access.
- Scanning craze for all kinds of ports – this will be never ending battle

# Routing Control Plane




- MD-5 authentication
  - Some deploy at customer's request
- Route filters limit what routes are believed from a valid peer
- Packet filters limit which systems can appear as a valid peer
- Limiting propagation of invalid routing information
  - Prefix filters
  - AS-PATH filters (trend is leaning towards this)
  - Route dampening (latest consensus is that it causes more harm than good)
- Not yet possible to validate whether legitimate peer has authority to send routing update

# Why Use Route Authentication

- Route Authentication equates to data origin authentication and data integrity
- In BGP, requires TCP resets to be authenticated so malicious person can't randomly send TCP resets
- In cases where routing information traverses shared networks, someone might be able to alter a packet or send a duplicate packet
- Routing protocols were not initially created with security in mind…..this needs to change….


# Hash Functions

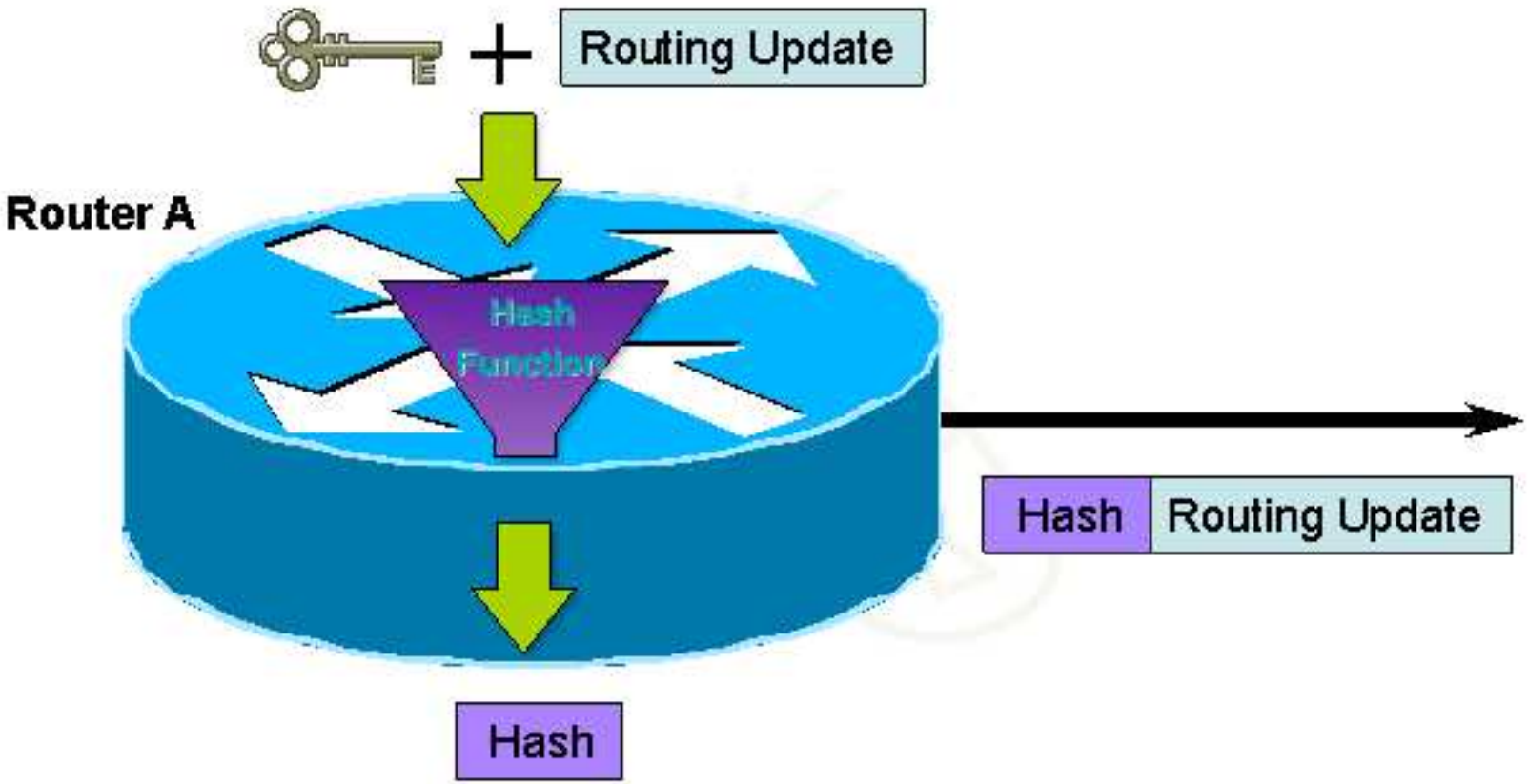
A *hash function* takes an input message of arbitrary length and outputs fixed-length code. The fixed-length output is called the *hash*, or the *message digest*, of the original input message.

Common Algorithms: MD-5 (128), SHA-1 (160)

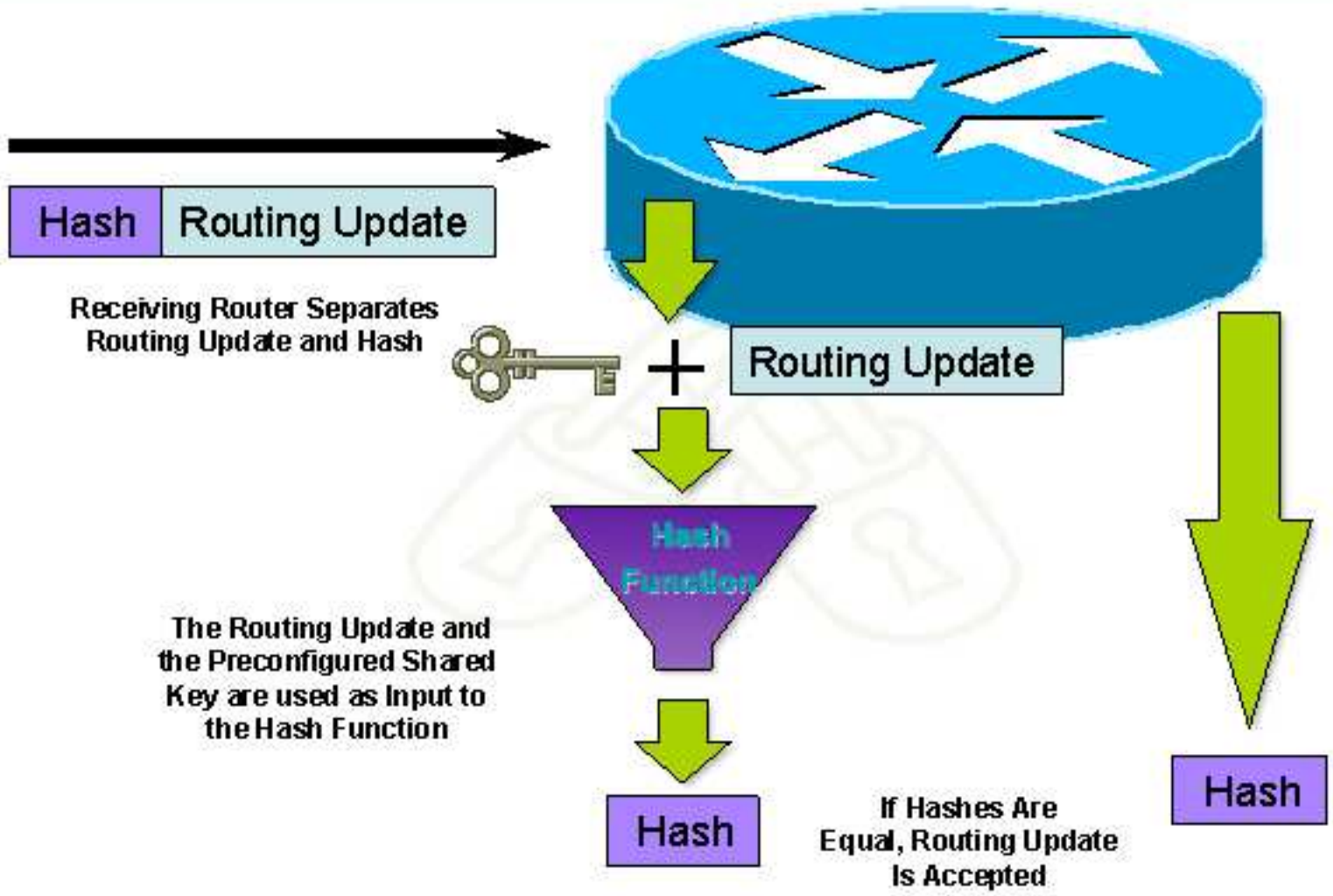
# Basics of Hash Algorithms

- Reduces a variable-length input to a fixed-length output
  - Output is called a *hash* or *message digest* or *fingerprint*
  - Output length is 128 bits for MD5 and 160 bits for SHA-1
- Requirements
  - Can't deduce input from output
  - Can't generate a given output
  - Can't find two inputs which produce the same output
- Used to
  - Create data checksum to detect data modification
  - Create fixed-length encryption keys from passwords

# MD-5 Based Authentication



# MD-5 Based Authentication



# Control Plane (Routing) Filters

- Filter traffic destined TO your core routers
- Develop list of required protocols that are sourced from outside your AS and access core routers
  - Example: eBGP peering, GRE, IPSec, etc.
  - Use classification filters as required
- Identify core address block(s)
  - This is the protected address space
  - Summarization is critical for simpler and shorter filter lists

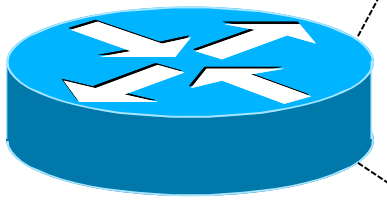
# BGP Prefix Filtering (same for IPv6 and IPv4 networks)

- All BGP Prefixes coming into your network and leaving your network need to be filtered to enforce a policy.
- The problem is most ISPs are not:
  - Filtering Comprehensively
  - Filtering their customer's prefixes
  - Filtering prefixes going out of their network.


# BGP IPv6 Prefix Filters To Consider

- Special-use prefixes
  - ::/0 exact : default route
  - ::1/128 : loop back address
  - ::/128 : unspecified address
  - ::/96 : IPv4-compatible IPv6 address
  - ::ffff:0:0/96 : IPv4-mapped IPv6 address
  - ::/8 or longer : reserved
  - fe80::/10 or longer : link-local address
  - fc00::/7 or longer : unique-local address
  - ff00::/8 or longer : multicast range (RFC3513)
  - fe00::/9 or longer : multicast range (RFC3513)
  - 2001:db8::/32or longer : documentation address
- Your own prefix
- The 6bone prefix (3ffe::/16)
- The 6to4 reserved address range (2002::/16) if not supporting 6to4 services (i.e. relays) and not providing transit services
- The Teredo address range (2001::/32) if not running a Teredo relay or offering a Teredo transit service

# Simple IPv6 Bogon Prefix Filter Example

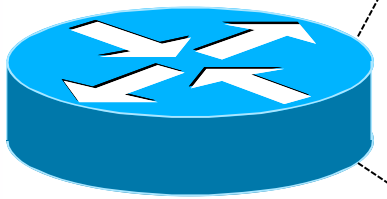


```
ipv6 prefix-list ipv6-special-use-pfx deny 0::/0 le 128
ipv6 prefix-list ipv6-special-use-pfx deny 0::1/128 le 128
ipv6 prefix-list ipv6-special-use-pfx deny 0::/128
ipv6 prefix-list ipv6-special-use-pfx deny 0::/96
ipv6 prefix-list ipv6-special-use-pfx deny 0::ffff:0:0/96
ipv6 prefix-list ipv6-special-use-pfx deny 0::/8 le 128
ipv6 prefix-list ipv6-special-use-pfx deny fe80::/10 le 128
ipv6 prefix-list ipv6-special-use-pfx deny fc00::/7 le 128
ipv6 prefix-list ipv6-special-use-pfx deny fe00::/9 le 128
ipv6 prefix-list ipv6-special-use-pfx deny ff00::/8 le 128
ipv6 prefix-list ipv6-special-use-pfx deny 2001:db8::/32 le 128
ipv6 prefix-list ipv6-special-use-pfx deny 3ffe::/16 le 128
```

# BGP Prefix Filters (RIR Allocations)

- APNIC
  - ftp://ftp.apnic.net/stats/apnic/delegated-apnic-latest
- RIPE NCC
  - ftp://ftp.ripe.net/pub/stats/ripencc/delegated-ripencc-latest
- ARIN
  - ftp://ftp.arin.net/pub/stats/arin/delegated-arin-latest
- LACNIC
  - ftp://ftp.lacnic.net/pub/stats/lacnic/delegated-lacnic-latest
- AfriNIC
  - ftp://ftp.afrinic.net/pub/stats/afrinic/delegated-afrinic-latest

# IPv6 RIR Allocation Prefix Filter Example (Needs Constant Updating)



```
ipv6 prefix-list ipv6-RIR-allocations-pfx permit 2001:0500::/30  ge 48 le 48
ipv6 prefix-list ipv6-RIR-allocations-pfx permit 2001:0678::/29  ge 48 le 48
ipv6 prefix-list ipv6-RIR-allocations-pfx permit 2001::/16 ge 35 le 35
ipv6 prefix-list ipv6-RIR-allocations-pfx permit 2001::/16 ge 19 le 32
ipv6 prefix-list ipv6-RIR-allocations-pfx permit 2003::/18 ge 19 le 32
ipv6 prefix-list ipv6-RIR-allocations-pfx permit 2400::/12 ge 13 le 32
ipv6 prefix-list ipv6-RIR-allocations-pfx permit 2600::/12 ge 13 le 32
ipv6 prefix-list ipv6-RIR-allocations-pfx permit 2610::/23 ge 24 le 32
ipv6 prefix-list ipv6-RIR-allocations-pfx permit 2620::/23 ge 40 le 48
ipv6 prefix-list ipv6-RIR-allocations-pfx permit 2800::/12 ge 13 le 32
ipv6 prefix-list ipv6-RIR-allocations-pfx permit 2A00::/12 ge 13 le 32
ipv6 prefix-list ipv6-RIR-allocations-pfx permit 2C00::/12 ge 13 le 32
ipv6 prefix-list ipv6-RIR-allocations-pfx permit 2001:0DF0::/29 ge 40 le 48
ipv6 prefix-list ipv6-RIR-allocations-pfx permit 2001:43F8::/29 ge 40 le 48
```

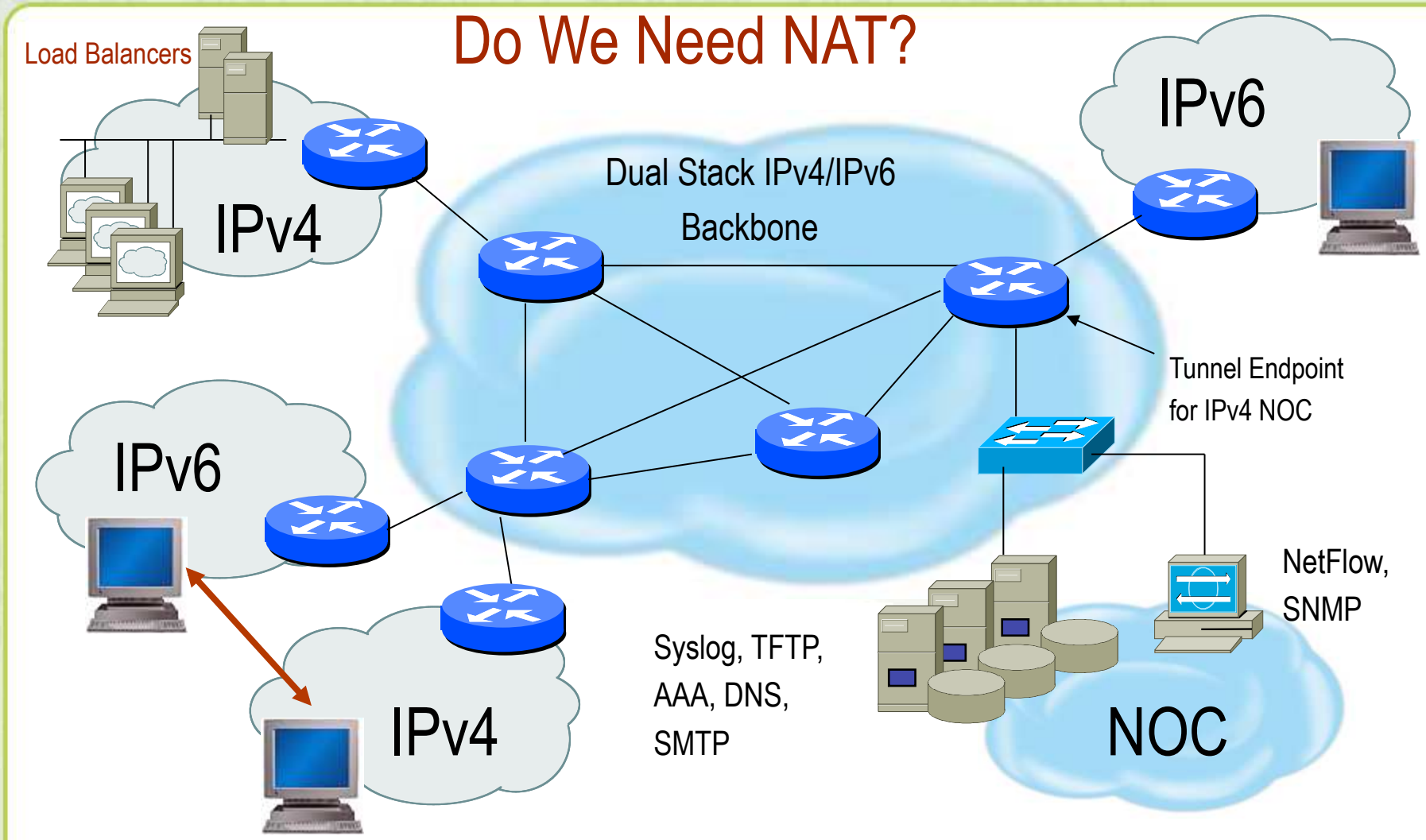
# Prefix Filter Bogons and RIR Blocks

- Templates available from the Bogon Project:
  - http://www.cymru.com/Bogons/index.html
- Cisco Template
  - ftp://ftp-eng.cisco.com/cons/isp/security/Ingress-Prefix-Filter-Templates/
- Juniper Template
  - http://www.qorbit.net/documents.html


# IPv6 Tunneling Considerations

- Manually configured tunnels are not scalable
- Automated tunnels require more diligence to provide effective security services
- Look at IETF Softwire Working Group
  - http://www.ietf.org/html.charters/softwire-charter.html
  - RFC 5619 (softwire-security-requirements)
- Deployments of 6to4, ISATAP and Teredo all require layered security models
  - Perform ingress firewall sanity checks
  - Log and audit tunneled traffic
  - Provide authentication where possible
  - Use IPsec where appropriate

# Network Address Translation



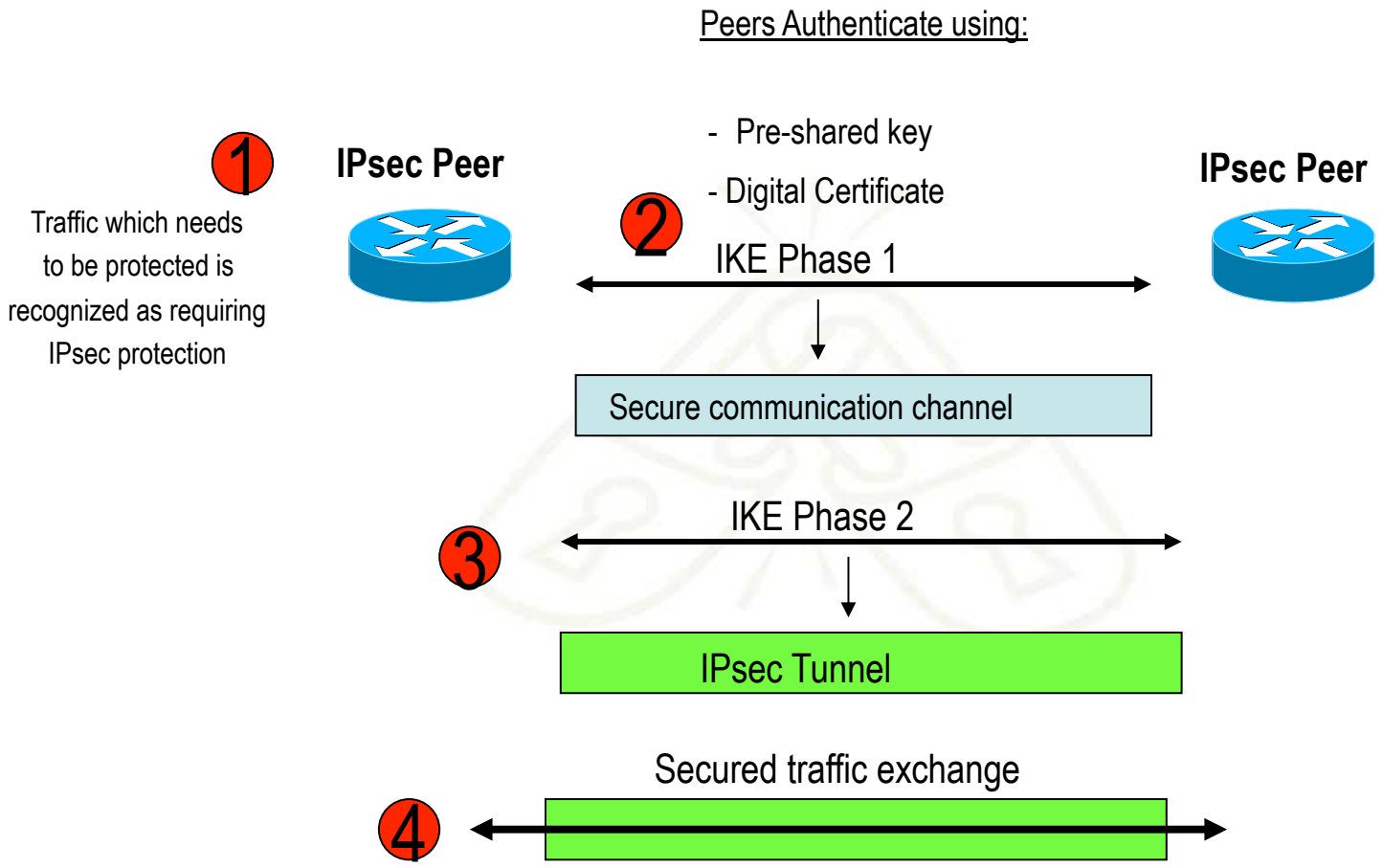
# IPsec in IPv6 Environments

- Bootstrapping credentials
  - Ship all devices with some embedded certificates and trusted roots
- Where useful
  - BGP/OSPFv3/ISIS Authentication
  - Syslogv6 / Radius (server-to-router)
  - TFTP / SNMP / Netflow
- Interoperable defaults
  - Have until widespread deployment of IPv6
  - Window of opportunity closing


# IPsec Components

- AH (Authentication Header)
  - Authentication is applied to the entire packet, with the mutable fields in the IP header zeroed out
  - If both ESP and AH are applied to a packet, AH follows ESP
  - Standard requires HMAC-MD5-96 and HMAC-SHA1-96….older implementations also support keyed MD5
- ESP (Encapsulating Security Payload)
  - Must encrypt and/or authenticate in each packet
  - Encryption occurs before authentication
  - Authentication is applied to data in the IPsec header as well as the data contained as payload
  - Standard requires DES 56-bit CBC and Triple DES. Can also use RC5, IDEA, Blowfish, CAST, RC4, NULL
- IKE (Internet Key Exchange)
  - Automated SA (Security Association) creation and key management

# IPsec with IKE



# IPsec IKE Phase 1 Uses DH Exchange

- First public key algorithm (1976)
- Diffie Hellman is a key establishment algorithm
  - Two parties in a DF exchange can generate a shared secret
  - There can even be N-party DF changes where N peers can all establish the same secret key
- Diffie Hellman can be done over an insecure channel
- IKE authenticates a Diffie-Hellman exchange
  - Pre-shared secret
  - Nonce (RSA signature)
  - Digital signature

# IKE Phase 1 Main Mode


1 **Negotiate IKE Policy**

IKE Message 1 (SA proposal) →

← IKE Message 2 (accepted SA)

2 **Authenticated DH Exchange**

IKE Message 3 (DH public value, nonce) →

← IKE Message 4 (DH public value, nonce)

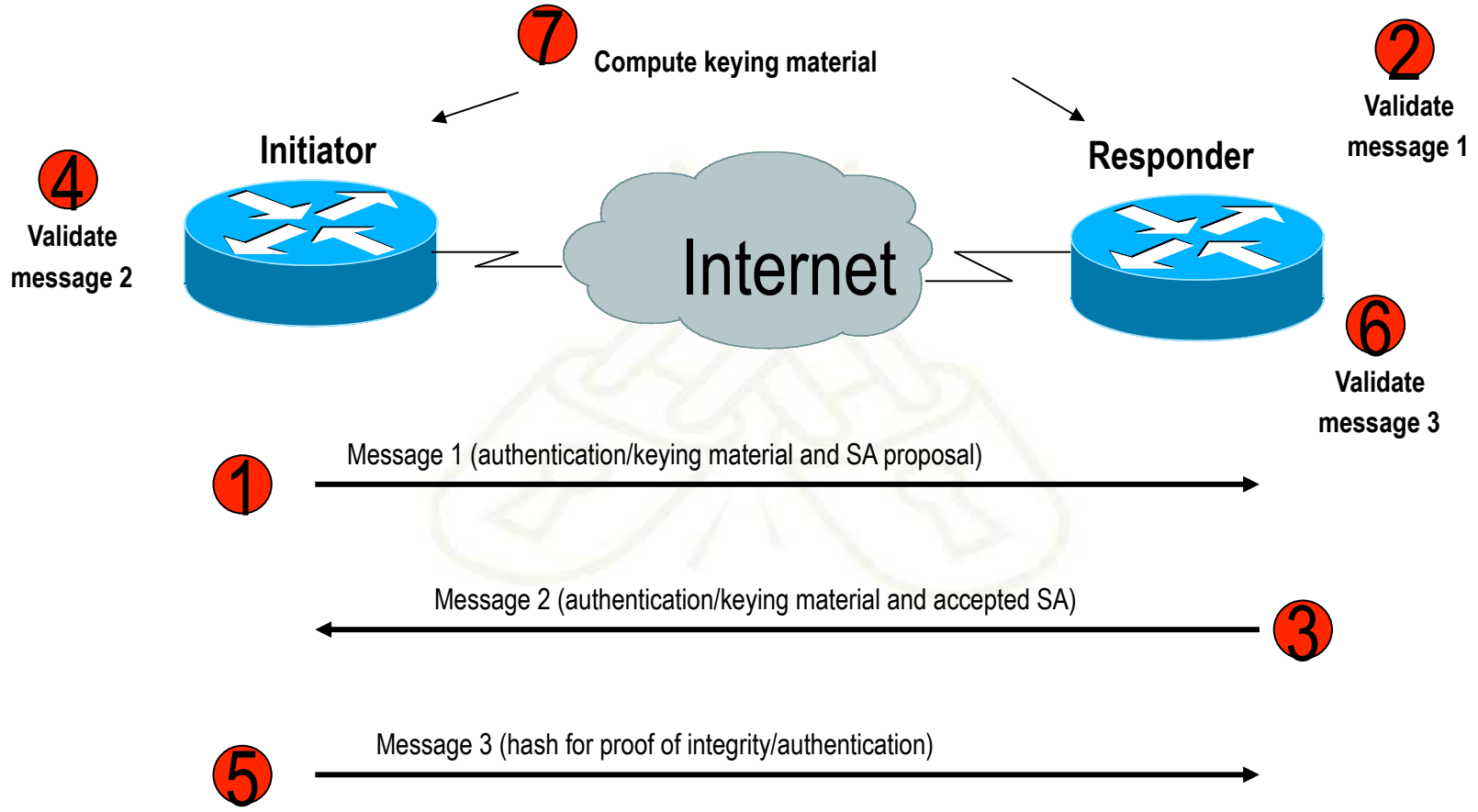
4 **Protect IKE Peer Identity**

IKE Message 5 (Authentication material, ID) →

← IKE Message 6 (Authentication material, ID)

(Encrypted)

# IKE Phase 2 Quick Mode



# Relevant Standard(s)

- IETF specific
  - rfc2409: IKEv1
  - rfc4301: IPsec Architecture (updated)
  - rfc4303: IPsec ESP (updated)
  - rfc4306: IKEv2
  - rfc4718: IKEv2 Clarifications
  - rfc4945: IPsec PKI Profile
- IPv6 and IPsec
  - rfc4294: IPv6 Node Requirements
  - Rfc4552: Authentication/Confidentiality for OSPFv3
  - rfc4877: Mobile IPv6 Using IPsec (updated)
  - rfc4891: Using IPsec to secure IPv6-in-IPv4 Tunnels

# Considerations For Using IPsec

- Security Services
  - Data origin authentication
  - Data integrity
  - Replay protection
  - Confidentiality
- Size of network
- How trusted are end hosts – can apriori communication policies be created?
- Vendor support
- What other mechanisms can accomplish similar attack risk mitigation

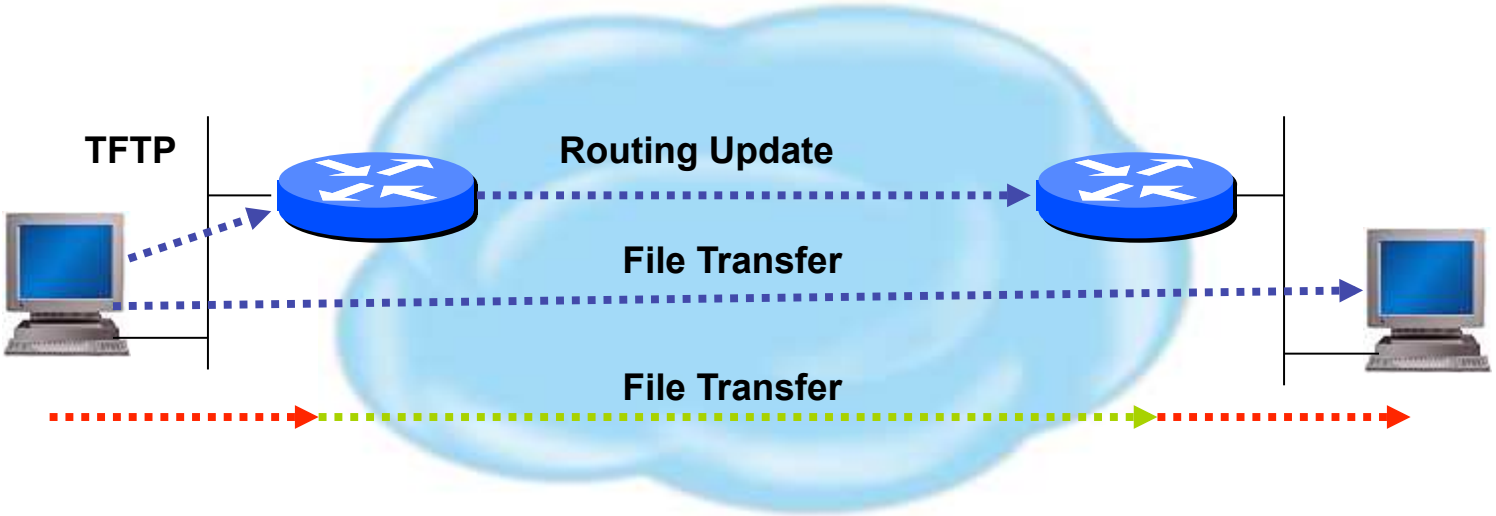
# Non-Vendor Specific Deployment Issues

- Historical Perception
  - Configuration nightmare
  - Not interoperable
- Performance Perception
  - Need empirical data
  - Where is the real performance hit?
- Standards Need Cohesion
- IPv6 Certification Entities Need Cohesion


# Vendor Specific Deployment Issues

- Lack of interoperable defaults
  - A default does NOT mandate a specific security policy
  - Defaults can be modified by end users
- Configuration complexity
  - Too many knobs
  - Vendor-specific terminology
- Good News: IPv6 support in most current implementations

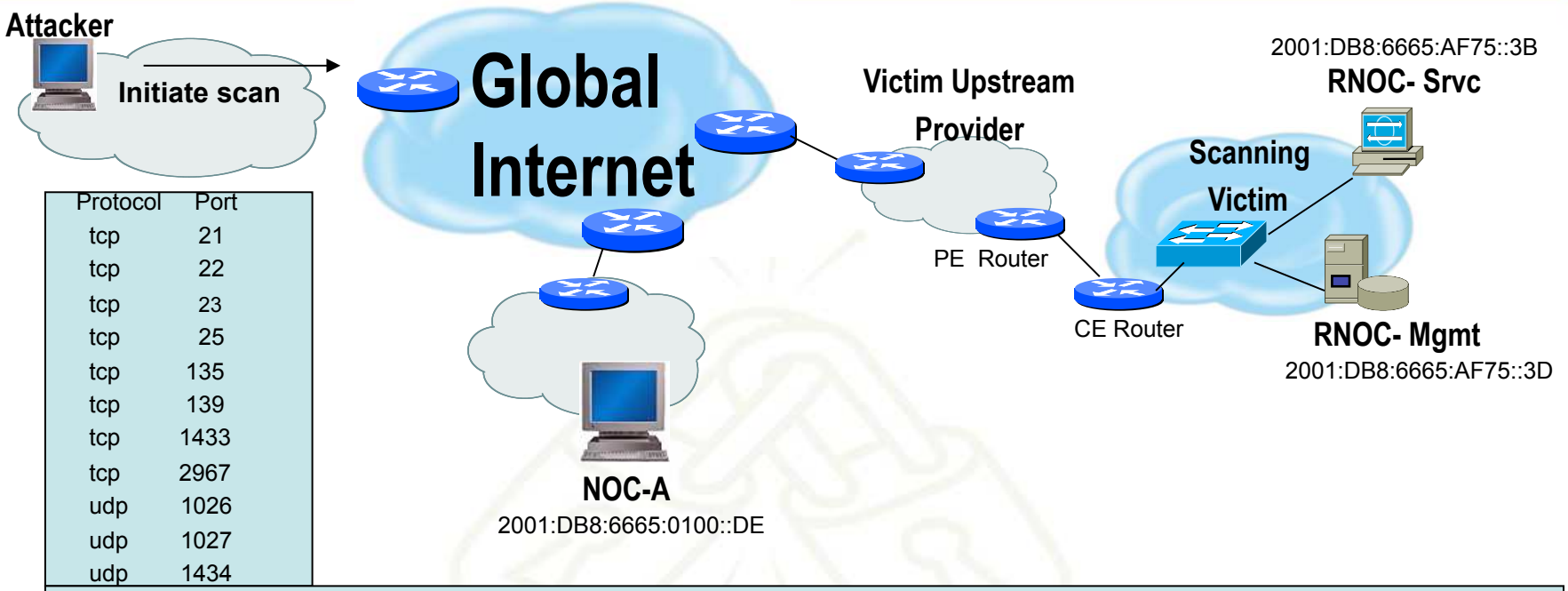
# Transport vs Tunnel Mode




**Transport Mode**: End systems are the initiator and recipient of protected traffic

**Tunnel Mode**: Gateways act on behalf of hosts to protect traffic

# Protecting Against Scanning Attacks

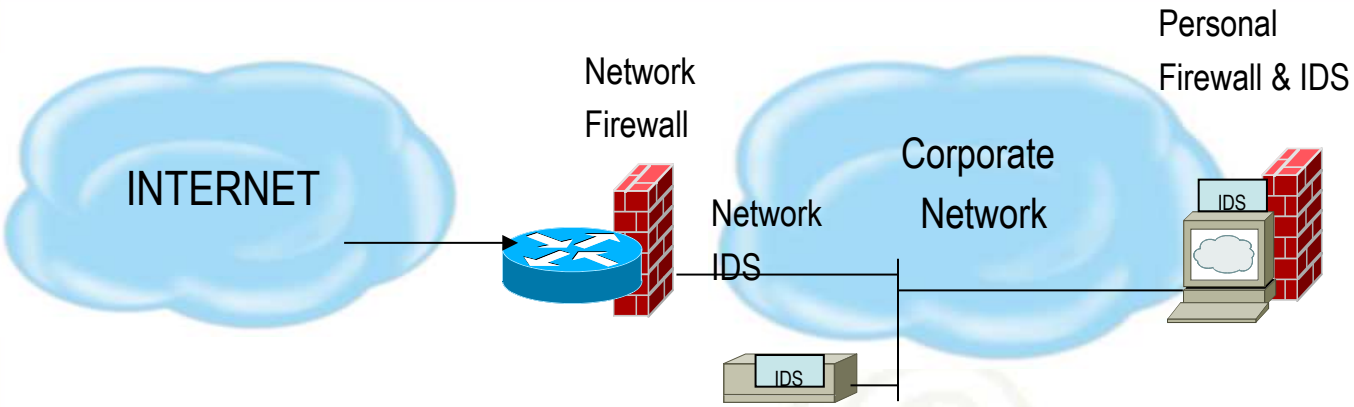


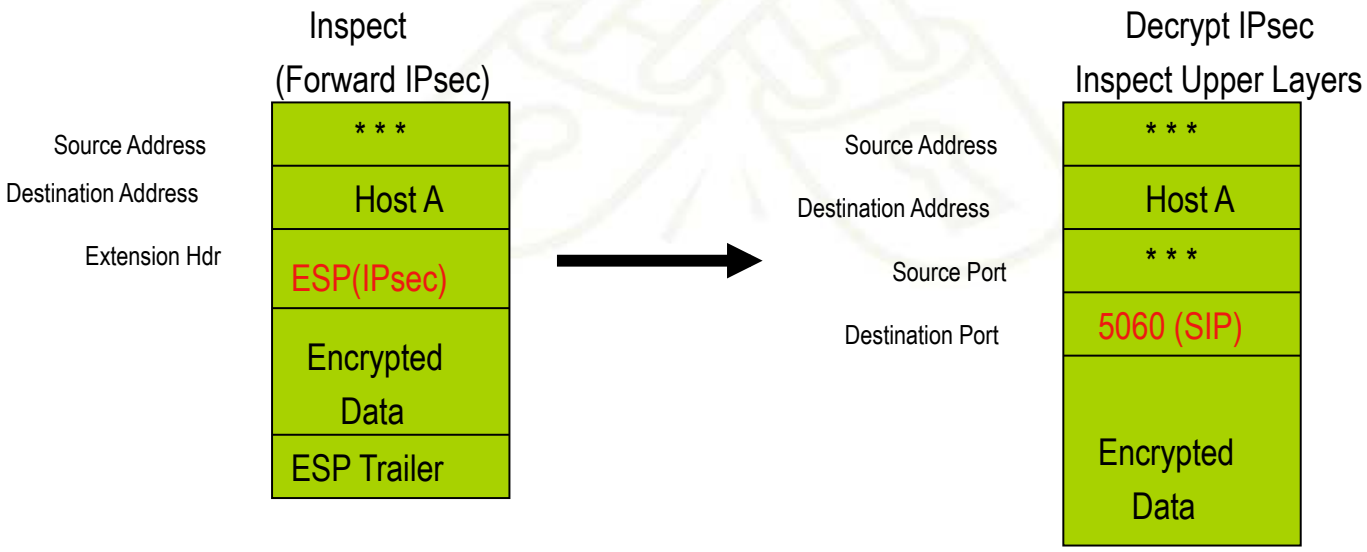

| Protocol | Port |
|---|---|
| tcp | 21 |
| tcp | 22 |
| tcp | 23 |
| tcp | 25 |
| tcp | 135 |
| tcp | 139 |
| tcp | 1433 |
| tcp | 2967 |
| udp | 1026 |
| udp | 1027 |
| udp | 1434 |

**IPsec Security Policy Database**

| From | To | Protocol | Dst Port | Policy |
|---|---|---|---|---|
| 2001:DB8:6665:0100::DE | 2001:DB8:6665:01C8::3B | TCP / UDP | 53 (DNS) | ESP: SHA1, AES-256 |
| 2001:DB8:6665:0100::DE | 2001:DB8:6665:AF75::3B | TCP | 25 (SNMP) | ESP: SHA1, AES-256 |
| 2001:DB8:6665:0100::DE | 2001:DB8:6665:AF75::3D | UDP | 1812/1813 (RADIUS) | ESP: SHA1, AES-128 |
| 2001:DB8:6665:0100::DE | 2001:DB8:6665:AF75::3D | UDP | 514 (Syslog) | ESP: SHA1, 3DES |
| 2001:DB8:6665:0100::DE | 2001:DB8:6665:AF75::/48 | TCP / UDP | ANY | ESP: SHA1 |

# Distributed IDS & Firewalls



# IPv6 IPsec Concerns

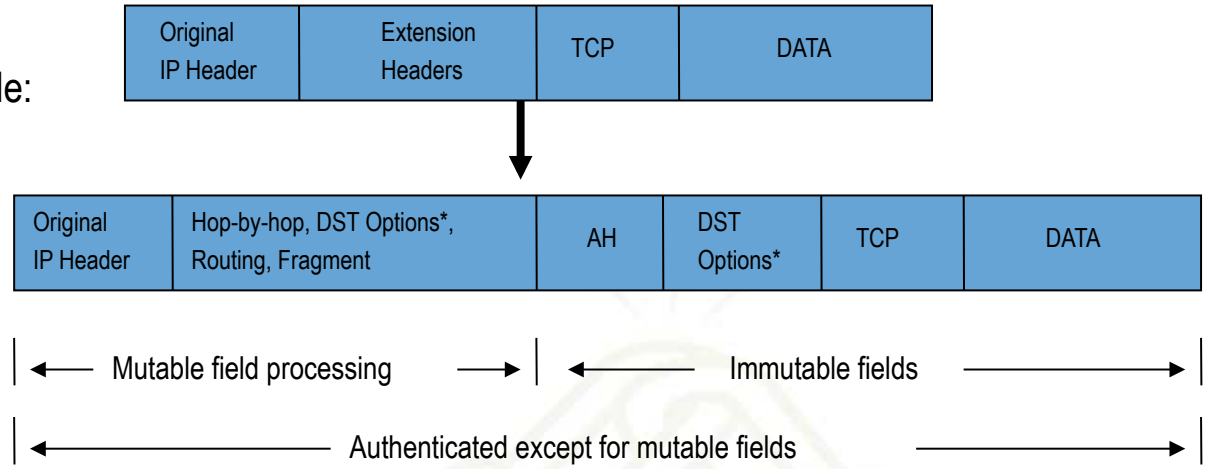
- Are enough people aware that IKEv2 is not backwards compatible with IKEv1?
  - IKEv1 is used in most IPv6 IPsec implementations
  - Will IKEv2 implementations first try IKEv2 and then revert to IKEv1?
- Is IPsec implemented for IPv6?
  - Some implementations ship IPv6 capable devices without IPsec capability….this needs to change
- OSPFv3
  - All vendors 'IF' they implement IPsec used AH
  - Latest standard to describe how to use IPsec says MUST use ESP w/Null encryption and MAY use AH

# IPv6 IPsec Concerns (cont)

- What is transport mode interoperability status?
  - Will end user authentication be interoperable?
- PKI Issues
  - Which certificates do you trust?
  - How does IKEv1 and/or IKEv2 handle proposals with certificates?
  - Should common trusted roots be shipped by default?
  - Who is following and implementing pki4ipsec-ikecert-profile (rfc4945)
- Have mobility scenarios been tested?
  - Mobility standards rely heavily on IKEv2
- ESP – how determine if ESP-Null vs Encrypted

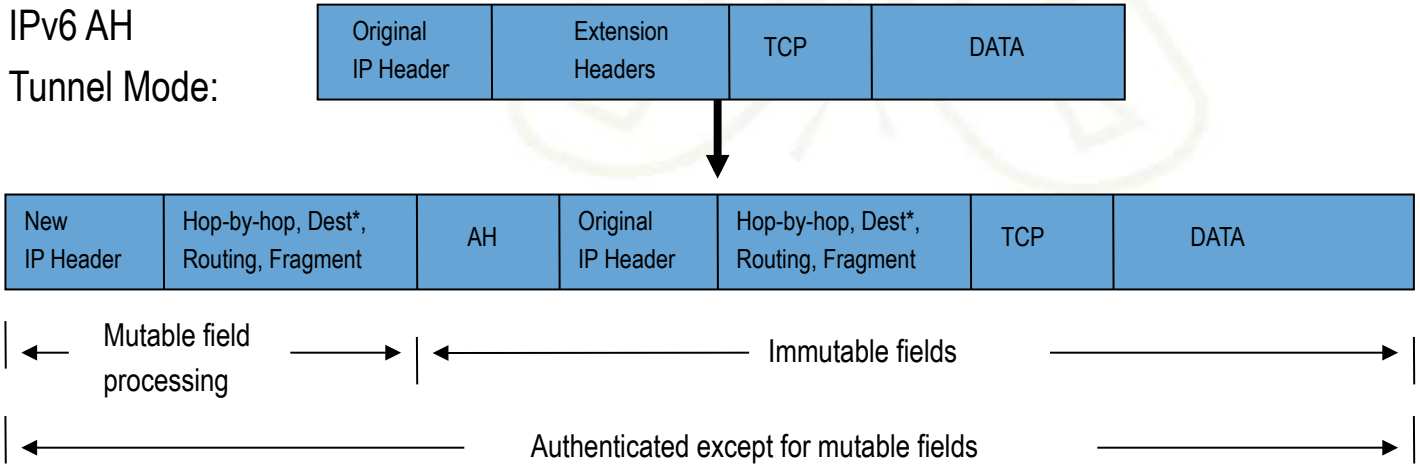
# IPv6 IPsec AH

IPv6 AH
Transport Mode:




Mutable Fields:
- DSCP
- ECN
- Flow Label
- Hop Limit

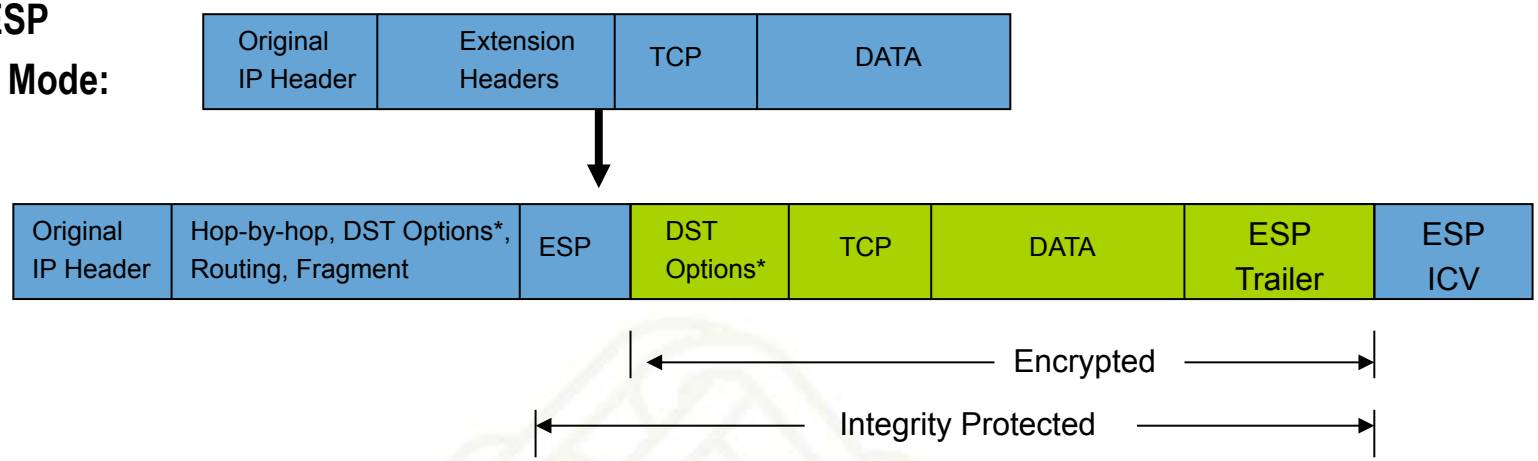
IPv6 AH
Tunnel Mode:



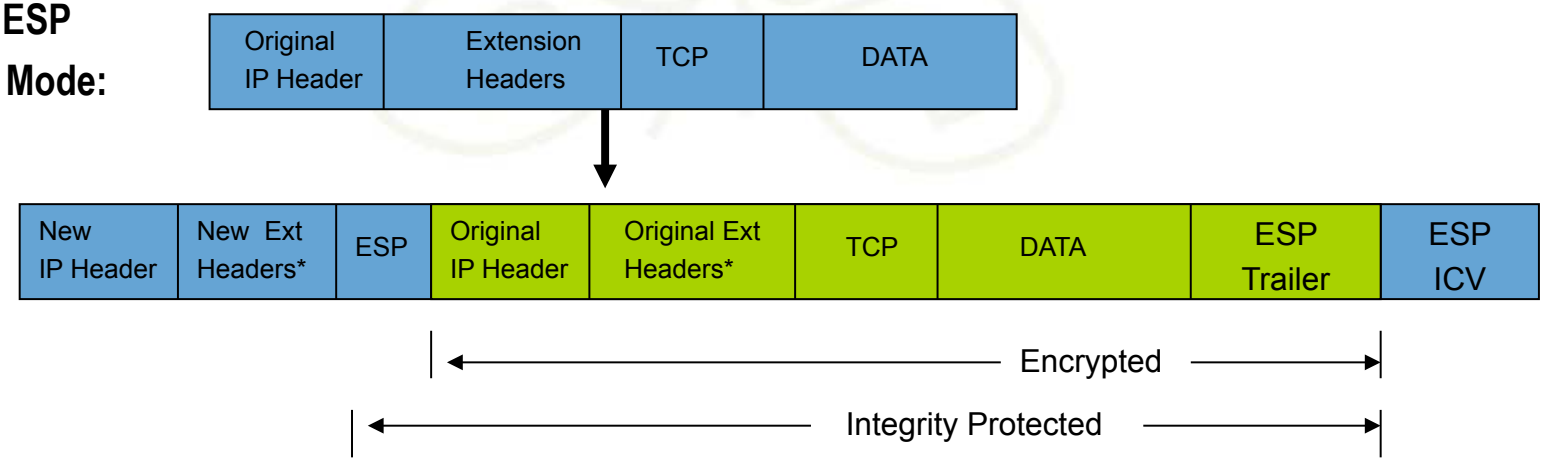

Mutable Fields:
- DSCP
- ECN
- Flow Label
- Hop Limit

# IPv6 IPsec ESP

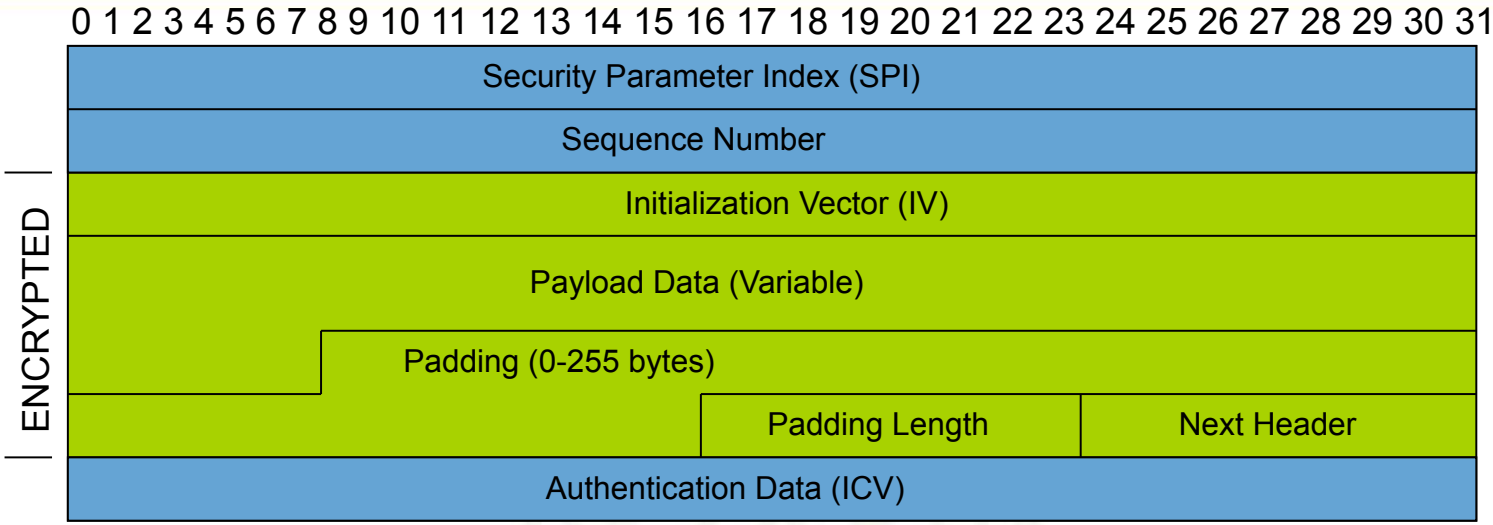
**IPv6 ESP Transport Mode:**




**IPv6 ESP Tunnel Mode:**



# ESP Header Format




**SPI:** Arbitrary 32-bit number that specifies SA to the receiving device

**Seq #:** Start at 1 and must never repeat; receiver may choose to ignore

**IV:** Used to initialize CBC mode of an encryption algorithm

**Payload Data:** Encrypted IP header, TCP or UDP header and data

**Padding:** Used for encryption algorithms which operate in CBC mode

**Padding Length:** Number of bytes added to the data stream (may be 0)

**Next Header:** The type of protocol from the original header which appears in the encrypted part of the packet

**Auth Data:** ICV is a digital signature over the packet and it varies in length depending on the algorithm used (SHA-1, MD5)


# Default Issues

## Vendor A

IKE Phase 1
- SHA1
- RSA-SIG
- Group 1
- Lifetime 86400 Sec
- Main Mode

IKE Phase 2
- PFS
- Group 1

## Vendor B

IKE Phase 1
- MD5
- Pre-Share Key
- Group 5
- Lifetime 86400 Sec
- Main Mode

IKE Phase 2
- PFS
- Group 5

## Vendor C

IKE Phase 1
- SHA1
- Pre-Share Key
- Group 2
- Lifetime 86400 Sec
- Aggressive Mode

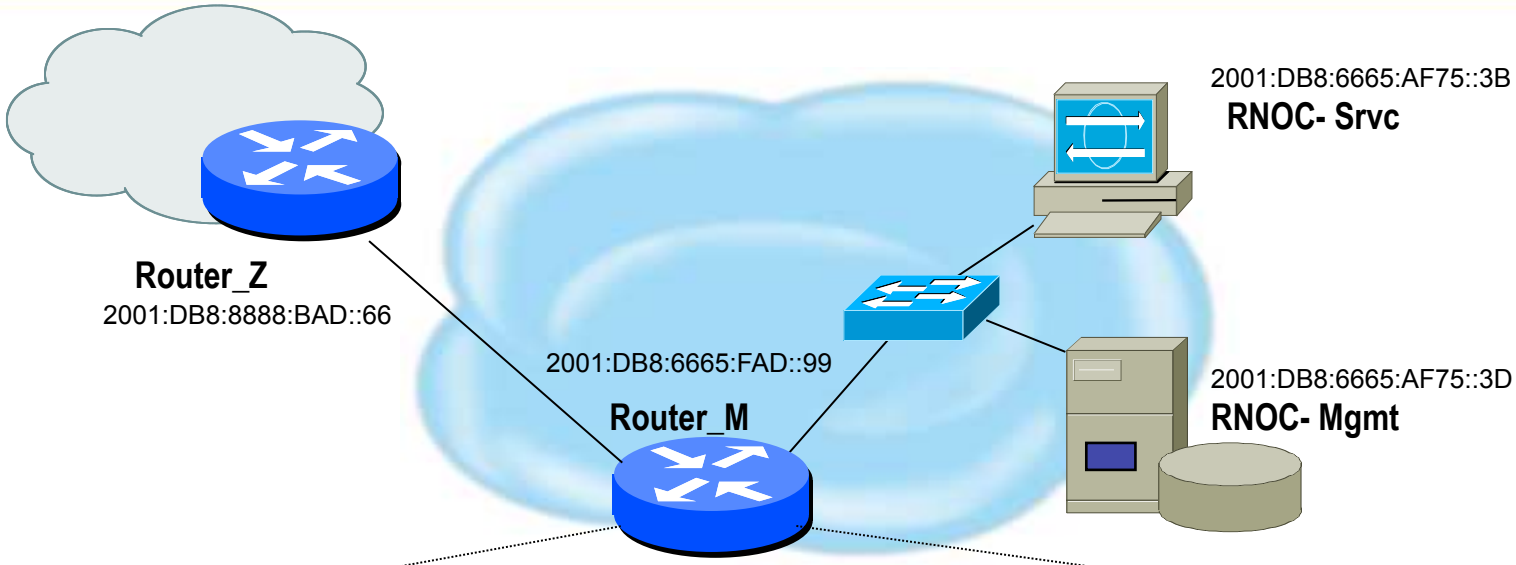
IKE Phase 2
- PFS
- Group 2

# Terminology Issues

| IKE Phase 1 | DH Key Length | IKE Phase 2 |
|---|---|---|
| IKE Phase 1 SA | DH Group | IKE Phase 2 SA |
| IKE SA | Modp # | IPsec SA |
| ISAKMP SA | Group # | Quick Mode |
| Main Mode | | |

Configuration complexity increased with vendor specific configuration terms

# Potentially Easy Configuration




Syslog server 2001:DB8:6665:AF75::3D authenticate esp-null sha1 pre-share ‘secret4syslog’

TFTP server 2001:DB8:6665:AF75::3D authenticate esp-null aes128 pre-share ‘secret4tftp’

BGP peer 2001:DB8:8888:BAD::66 authenticate esp-null aes128 pre-share ‘secret4AS#XXX’

# Interoperable Defaults For SAs

- Security Association groups elements of a conversation together
  - ESP encryption algorithm and key(s)
  - Cryptographic synchronization
  - SA lifetime
  - SA source address
  - Mode (transport or tunnel)

**How Do We Communicate Securely ?**

Do we want integrity protection of data ?

Do we want to keep data confidential ?

Which algorithms do we use ?

What are the key lengths ?

When do we want to create new keys ?

Are we providing security end-to-end ?

# IPv6 IPsec WishList

- Common Terminology
- Interoperable Defaults
  - RFC 4308 was a good start but needs to be updated
- Interoperability Tests
  - Both transport and tunnel mode
  - Mobility scenarios
- API Standards
- Repeatable performance data

# Pretty Good IPsec Policy

- IKE Phase 1 (aka ISAKMP SA or IKE SA or Main Mode)
  - 3DES (AES-192 if both ends support it)
  - Lifetime (480 min = 28800 sec)
  - SHA-1
  - DH Group 14 (aka MODP# 14)
- IKE Phase 2 (aka IPsec SA or Quick Mode)
  - 3DES (AES-192 if both ends support it)
  - Lifetime (60 min = 3600 sec)
  - SHA-1
  - PFS 2
  - DH Group 14 (aka MODP# 14)


# Routers: Configuring IPsec

- For IPv6, consider using transport mode between routers and syslog servers, tftp servers, snmp servers, etc.
- Document for Cisco IPv6 IPsec configuration:
  - http://www.lseltd.com/univercd/cc/td/doc/product/software/ios123/123cgcr/ipv6_c/v6_ipsec.pdf
- Document for Juniper IPsec configuration:
  - http://www.pacificbroadband.com/techpubs/software/junos/junos83/feature-guide-83/html/fg-ipsec13.html#1139838

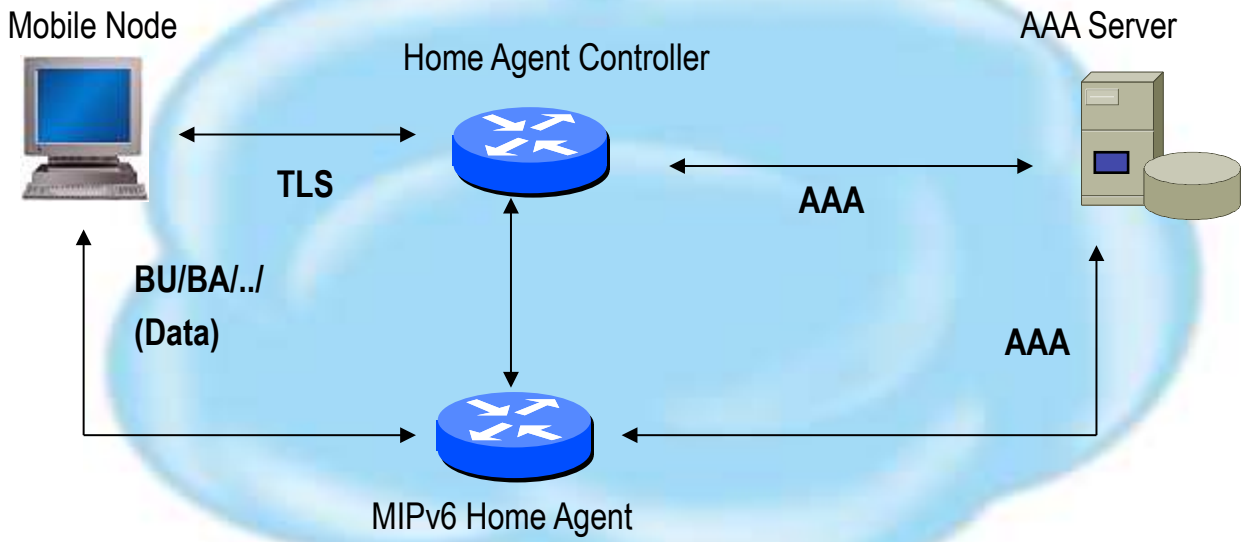
# Latest IETF Work related IPv6 Security

- CPE Device Issues / Concerns
  - draft-ietf-v6ops-cpe-simple-security-12.txt
  - draft-ietf-v6ops-ipv6-cpe-router-07.txt
- Router Advertisements
  - Draft-ietf-v6ops-rogue-ra-02.txt
  - draft-ietf-v6ops-ra-guard-08.txt
- SeND / CGI
  - draft-ietf-savi-send-03.txt
  - RFC 5909 (security ND proxy problem statement)
  - draft-ietf-csi-hash-threat-09
  - Draft-ietf-csi-proxy-send-04.txt
  - Draft-ietf-csi-send-cert-06.txt
  - Draft-ietf-csi-dhcpv6-cgs-ps-04.txt

# Latest IETF Work related IPv6 Security

- Tunneling Protocols
  - draft-gont-6man-teredo-loops-00.txt
  - draft-ietf-v6ops-tunnel-loops-00.txt
  - draft-ietf-v6ops-tunnel-security-concerns-02.txt
- General
  - draft-gont-6man-flowlabel-security-00.txt
  - draft-ietf-6man-node-req-bis-05
    - IPsec from 'MUST' to 'SHOULD'
- IPsec
  - RFC 5739 (IPv6 Configuration in IKEv2)

# Mobility and Security




MN and HAC signaling message exchange protected by TLS

Security Association between MN and HAC must not be tied to CoA of the MN

HAC, HA and AAA Server are logically separate entities that can be combined


# IPv6 Security Summary

- Don’t break existing IPv4 network
- Securing IPv6
  - Addressing infrastructure needs careful thought
  - Go native where possible to avoid tunnels being used for malicious behavior that’s hard to track
  - Use simple initial controls when getting started with IPv6
    - Vty access-lists
    - Sanity check filters on ingress/egress interfaces
- Do NOT blindly mimic IPv4 security architecture
  - Feature parity not necessarily what you want