

# Security Workshop

## MENOG 4 – Manama, Bahrain

### April 2009

Merike Kaeo

[merike@doubleshotsecurity.com](mailto:merike@doubleshotsecurity.com)



# Agenda

- What Are We Protecting Against?
- Proactive Mitigation Techniques
  - Securing The Device
  - Securing Data Traffic
  - Securing The Routing Infrastructure
  - Mitigating DDoS Attacks
- Auditing / Logging
  - Tools and Techniques
- New Paradigms



# What Are We Protecting Against?



Security Workshop – MENOG 4 Manama, Bahrain - April 2009

# Basic Terms

- **Threat**
  - Any circumstance or event with the potential to cause harm to a networked system
    - Denial of Service / Unauthorized Access / Impersonation / Worms / Viruses
- **Vulnerability**
  - A weakness in security procedures, network design, or implementation that can be exploited to violate a corporate security policy
    - software bugs / configuration mistakes / network design flaw
- **Risk**
  - The possibility that a particular vulnerability will be exploited
    - *Risk analysis*: The process of identifying security risks, determining their impact, and identifying areas requiring protection



# How Can The Threats Be Realized ?

- Protocol error
  - Routing protocol itself
  - TCP issues for BGP
- 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



# Passive vs Active Attacks

- Passive Attacks
  - Eavesdropping
  - Offline cryptographic attacks
- Active Attacks
  - Replay
  - Man-In-The-Middle
  - Message Insertion
  - Spoofing (device or user)
  - Denial of Service
  - Protocol specific attacks

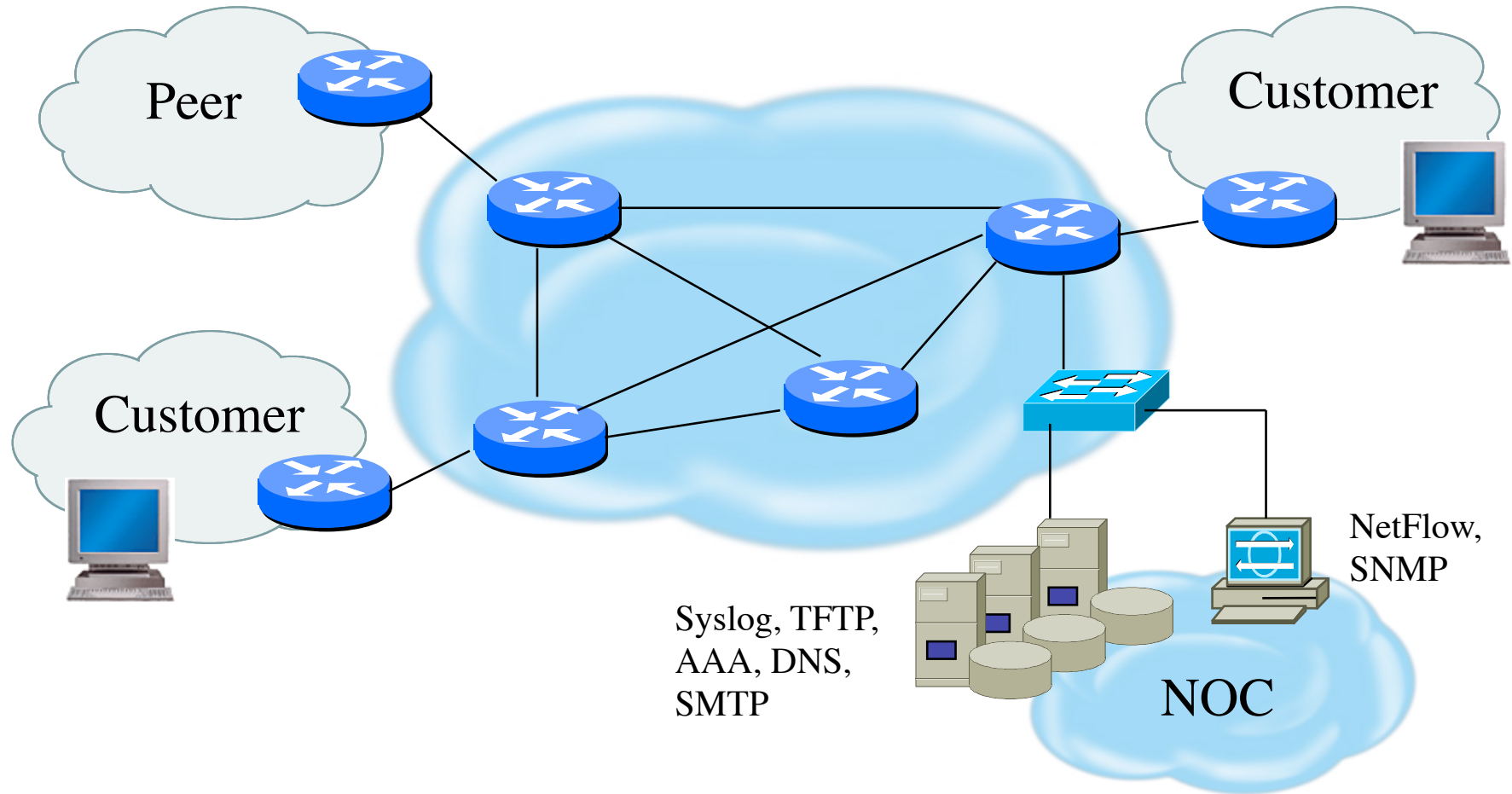


# What Can Intruders Do?

- 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



# Infrastructure Vulnerabilities





# What Can We Do To Protect The Infrastructure ?

- Understand the Problem (Risk Analysis)
- Establish an Effective Infrastructure Security Policy
  - physical security
  - logical security
  - control/management plane
  - routing plane
  - data plane
- Have Procedures In Place For Incident Response
  - procedures for assessing software vulnerability risk
  - auditing configuration modifications



# What Are Security Goals?

- Controlling Data Access
- Controlling Network Access
- Protecting Information in Transit
- Ensuring Network Availability
- Preventing Intrusions
- Responding To Incidences



# Security Properties

- Confidentiality
  - Access to information is restricted to those who are privileged to see it
- Integrity
  - Having trust that information has not been altered during its transit from sender to intended recipient
- Accountability
  - Non-repudiation: property of a cryptographic system that prevents a sender from denying later that he or she sent a message or performed a certain action
- Availability
  - Information or resources are accessible when required



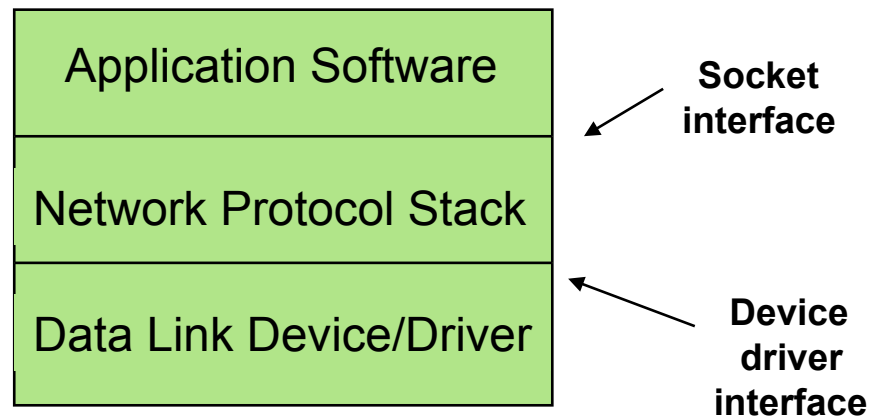
# Security Services

- **Authentication**
  - Process of verifying the claimed identity of a device, user and/or application
- **Authorization**
  - Rights and permissions granted to a user, device or application that enables access to resources
- **Access Control**
  - Means by which authorized user has access to resources
- **Encryption**
  - Mechanism by which information is kept confidential
- **Auditing**
  - Process that keeps track of networked activity



# Security

## Host / Network / Application



**Need to implement security solutions at all layers in a reasonable fashion.**

**So....Big Question: What Is Reasonable?**



# Risk Mitigation vs Cost

***Risk mitigation:*** the process of selecting appropriate controls to reduce risk to an acceptable level.

The ***level of acceptable risk*** is determined by comparing the risk of security hole exposure to the cost of implementing and enforcing the security policy.

***Assess the cost of certain losses and do not spend more to protect something than it is actually worth.***



# Traditional IT Security Policies

- Physical security controls
  - Media
  - Equipment location
  - Environmental safeguards
- Logical security controls
  - Subnet boundaries
  - Routing boundaries
  - Logical access control
- System and data integrity
  - Firewalls
  - Network services
- Data confidentiality
- Verify / Monitor / Audit
  - Accounting
  - Management
  - Intrusion detection



# Added Policy Considerations

- Policies and procedures for staff
  - Secure backups
  - Equipment certification
  - Use of Portable Tools
  - Audit Trails
  - Incident Handling
- Security awareness training for users of the network
  - Critical for all personnel
  - Added challenge of non-network savvy personnel





# Incident Handling

- You will have to deal with a security incident
- DON'T PANIC!! :)
- Systematically assess vulnerabilities and where to possibly place more effort on auditing / monitoring
- Detect / Assess / Respond
  - Automate as much as possible
  - Requires detailed operational guidance



# Closing Thoughts on Security Policy

- Can it be implemented technically?
- Are you able to implement it organizationally?
- Can you enforce it with security tools and/or sanctions?
- Does it clearly define areas of responsibility for the users, administrators, and management?
- Is it flexible and adaptable to changing environments?



# Proactive Mitigation Techniques



Security Workshop – MENOG 4 Manama, Bahrain - April 2009

# Securing The Device



Security Workshop – MENOG 4 Manama, Bahrain - April 2009

# Device Physical Access

- Equipment kept in highly restrictive environments
- Console access
  - password protected
  - access via OOB management
- Individual users authenticated
- Social engineering training and awareness



# Device Access (Cisco)

- 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

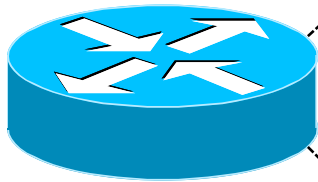


# Access Control Best Practices

- 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

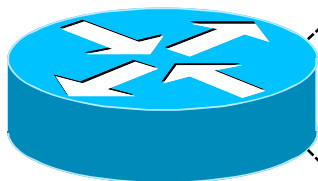


# Management Plane Filters

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



# Authenticate Individual Users



```
username merike secret merike-secret  
username gaurab secret gaurab-secret  
username pfs secret pfs-secret  
username staff secret group-secret
```

***Do NOT have group passwords!***



# Restrict Access To Trusted Hosts

- Use filters to specifically permit hosts to access an infrastructure device
- Example

```
access-list 103 permit tcp host 192.168.200.7 192.168.1.0 0.0.0.255 eq 22 log-input
access-list 103 permit tcp host 192.168.200.8 192.168.1.0 0.0.0.255 eq 22 log-input
access-list 103 permit tcp host 192.168.100.6 192.168.1.0 0.0.0.255 eq 23 log-input
access-list 103 deny ip any any log-input
!
line vty 0 4
access-class 103 in
transport input ssh telnet
```

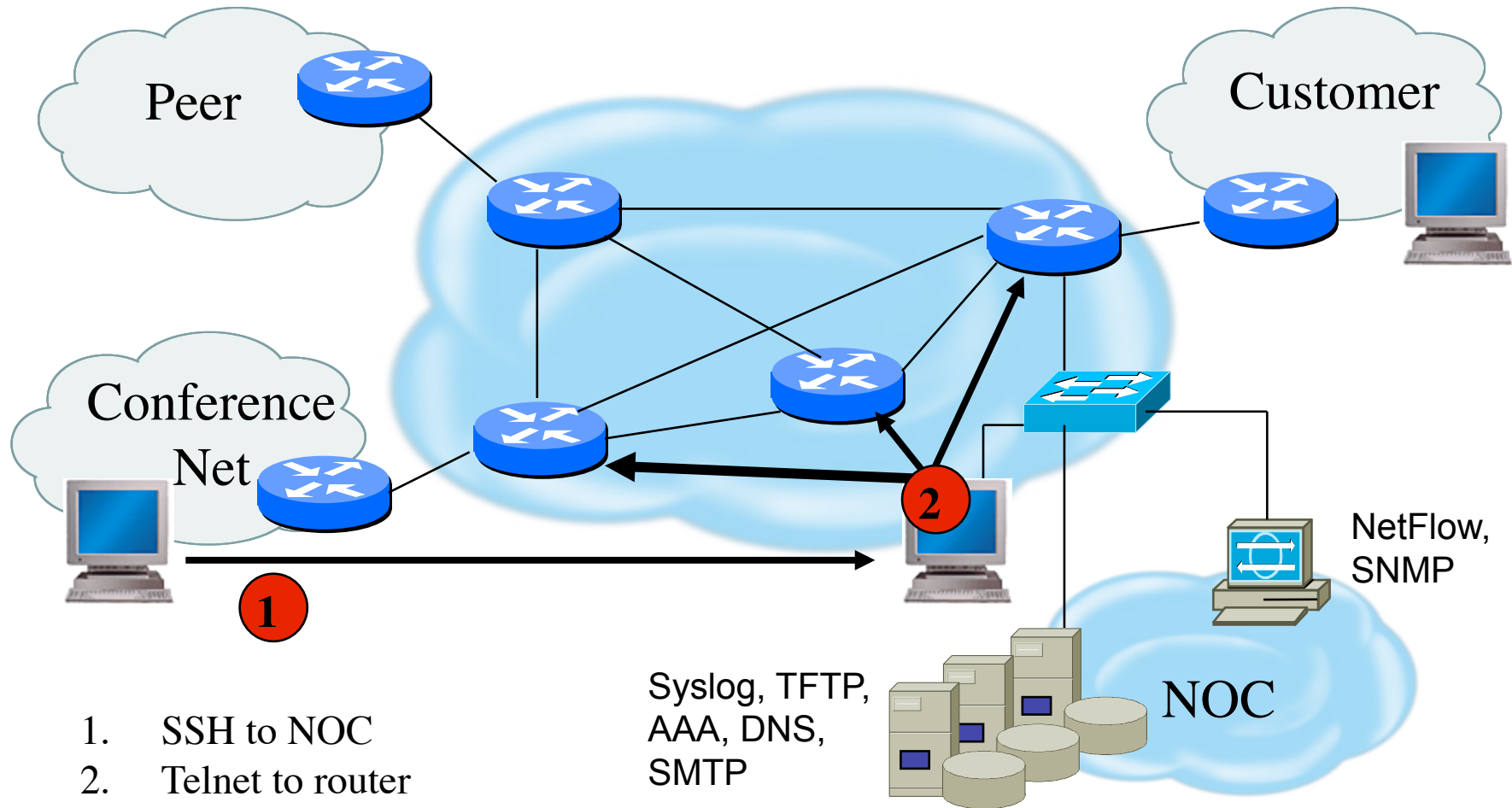


# Telnet is Insecure

- 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'



# 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](http://www.openssh.org)
  - Supports both ssh1 and ssh2
- Putty client for Windows
  - [www.chiark.greenend.org.uk/~sgtatham/putty/](http://www.chiark.greenend.org.uk/~sgtatham/putty/)





# SSH

## Enable SSH on cisco router

To enable SSH, you need to configure a hostname and a domainname for your router before generating the RSA key pair. Enter 1024 for the size of the key modulus when requested....this is the recommended minimum size

```
Router (config)# hostname <hostname>  
Router (config)# ip domain-name <domainname>  
Router (config)# crypto key generate rsa
```

## Allow SSH access via virtual terminals

```
line vty 0 4  
transport input ssh
```



# 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 f00bAr RO 8*  
*access-list 8 permit 127.1.3.5*
- Explicitly direct SNMP traffic to an authorized management station.  
*snmp-server host f00bAr 127.1.3.5*



# SNMP Best Practices

- 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

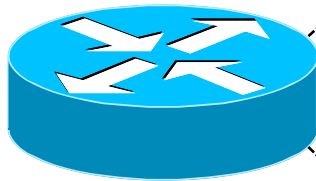


# Secure Logging Infrastructure

- 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
```

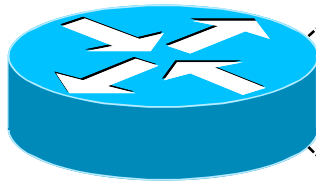
```
You should not be on this device.
```

```
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.



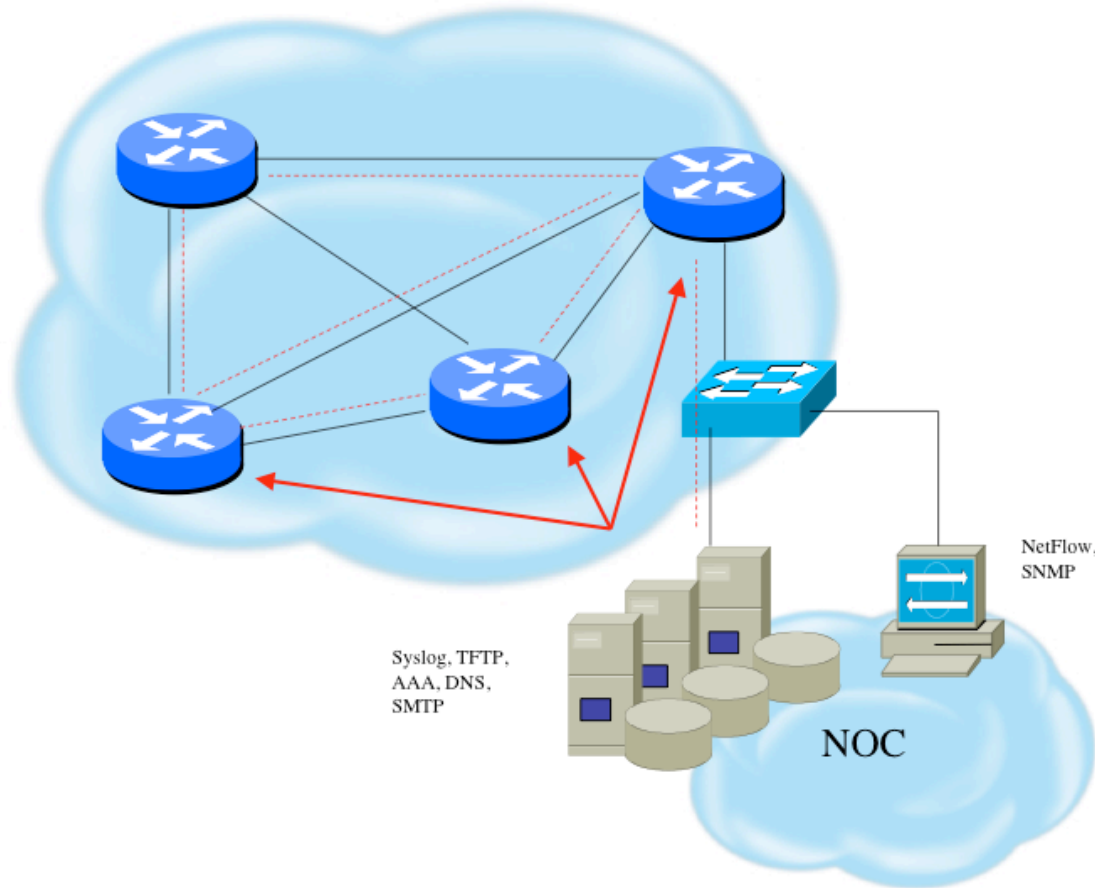
# Turn Off Unused Services

- 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





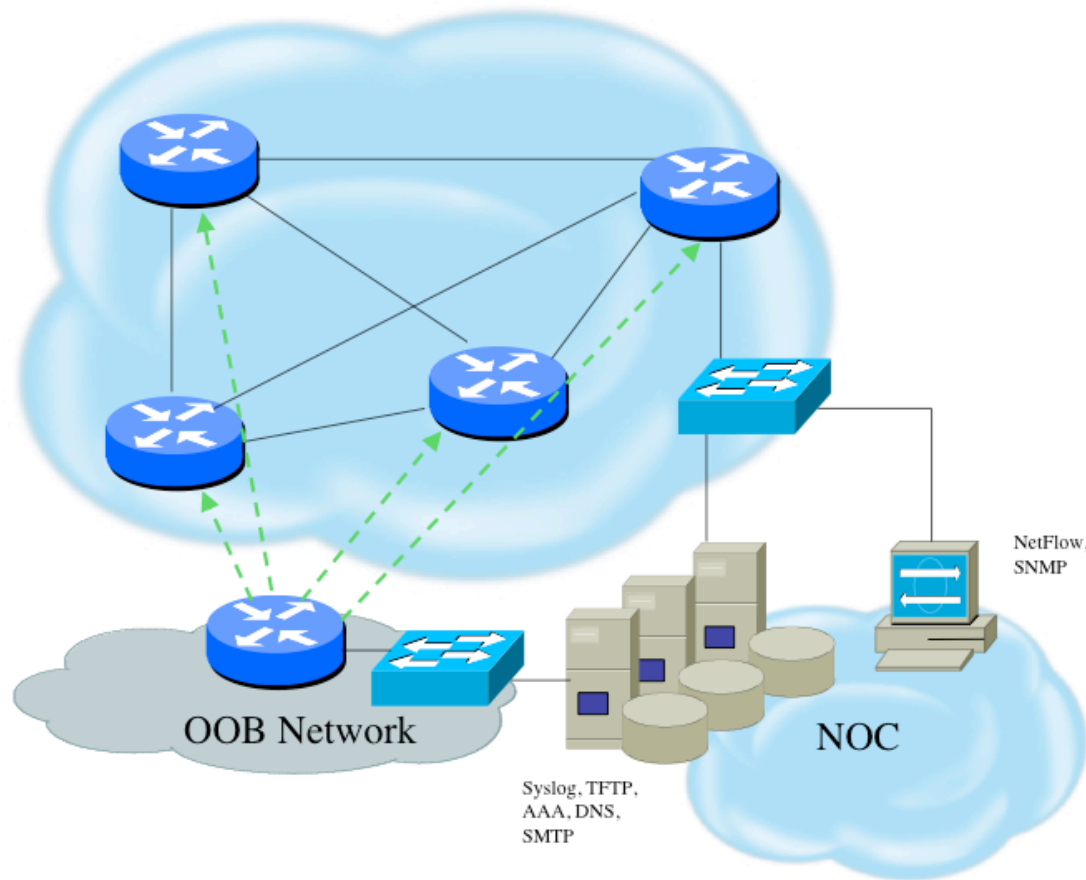
# Device In-Band Management



- 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 Common Practice

- SSH primarily used; Telnet only from jump hosts
- 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

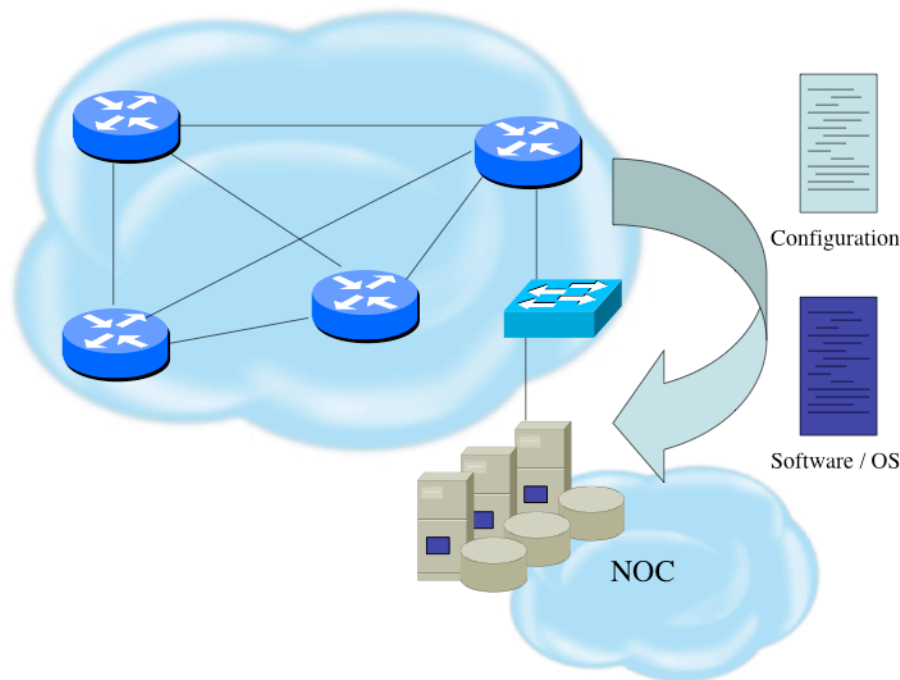


# System Images and Configuration Files

- Careful of sending configurations where people can snoop the wire
  - CRC or MD5 validation
  - Sanitize configuration files
- SCP should be used to copy files
  - TFTP and FTP should be avoided
- Use tools like 'rancid' to periodically check against modified configuration files



# Software Upgrade / Integrity



- Files stored on specific systems with limited access
- All access to these systems are authenticated and audited
- SCP is used where possible; FTP is NEVER used; TFTP still used
- Configuration files are polled and compared on an hourly basis
- Filters limit uploading / downloading of files to specific systems
- Many system binaries use MD-5 checks for integrity
- Configuration files are stored with obfuscated passwords



# Fundamental Device Protection Summary

- Secure logical access to routers with passwords and timeouts
- Never leave passwords in clear-text
- Authenticate individual users
- Restrict logical access to specified trusted hosts
- Allow remote vty access only through ssh
- Disable device access methods that are not used
- Protect SNMP if used
- Shut down unused interfaces
- Shut down unneeded services
- Ensure accurate timestamps for all logging
- Create appropriate banners
- Test device integrity on a regular basis

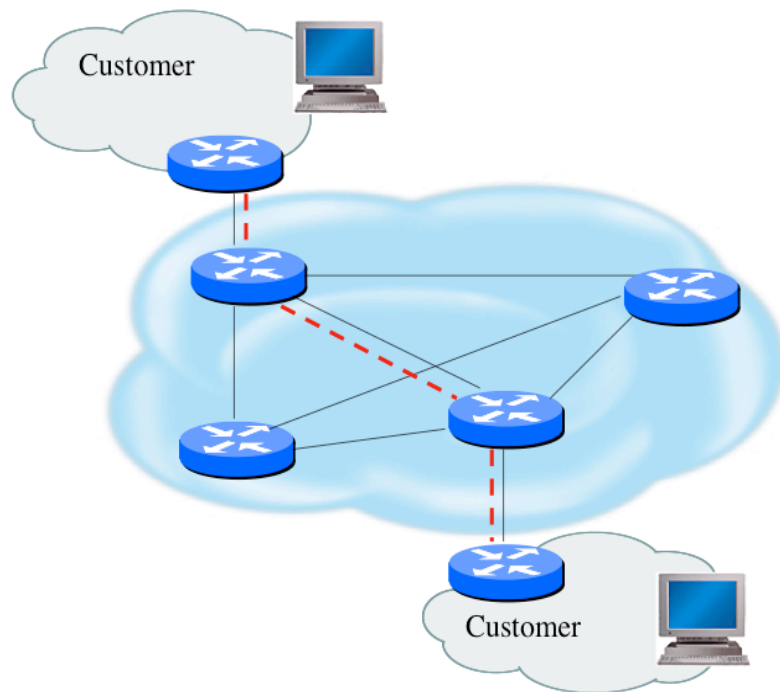


# Securing The Data



Security Workshop – MENOG 4 Manama, Bahrain - April 2009

# 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?



# Router Filter vs Standalone Firewall Tradeoffs

## *USING A ROUTER AS FIREWALL*

- Increased CPU cycles and memory usage
- Single device to maintain

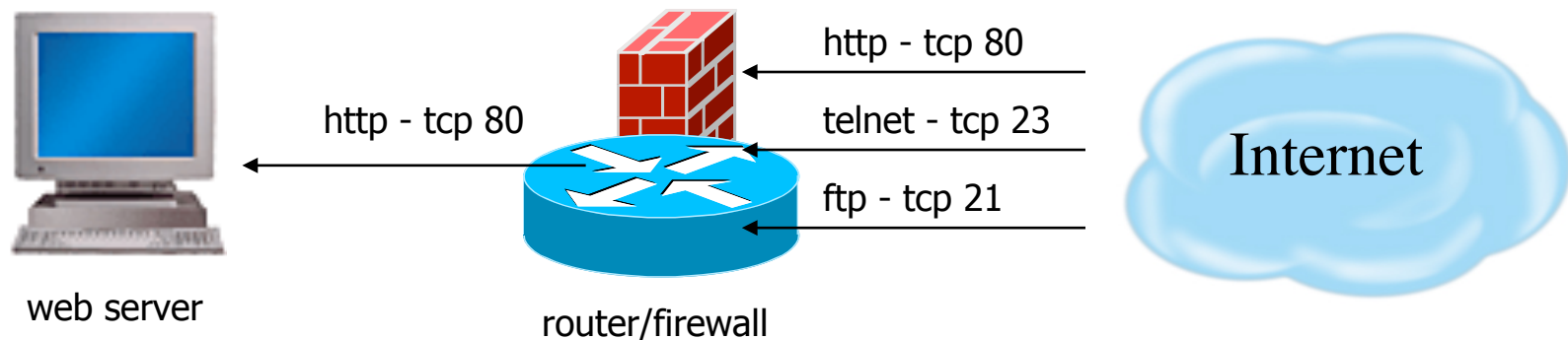
## *USING A STANDALONE FIREWALL*

- Additional hardware cost and maintenance
- Additional software purchase and updates
- Administrative setup and training
- Offload resources used from router



# Packet Filtering Firewall

- examines the source and destination address of the data packet and either allows or denies the packet from traveling the network
- blocks access through the firewall to any packets, which try to access ports which have been declared "off-limits"

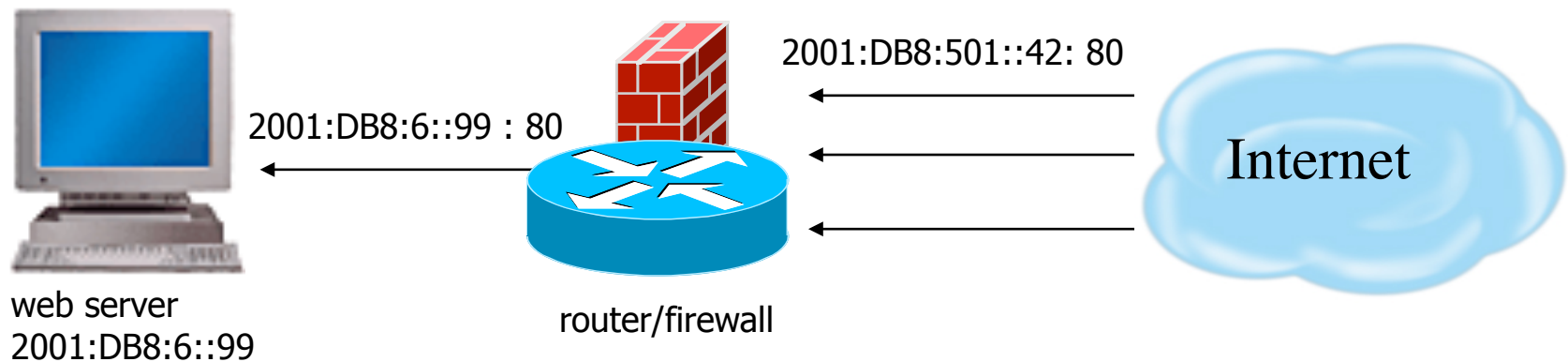


Allow only http - tcp 80  
Drop anything else



# Application Layer Firewall

- Also known proxy firewalls, application gateway
- attempts to hide the configuration of the network behind the firewall by acting on behalf of that network/servers
- All requests for access are translated at the firewall so that all packets are sent to and from the firewall, rather than from the hosts behind the firewall

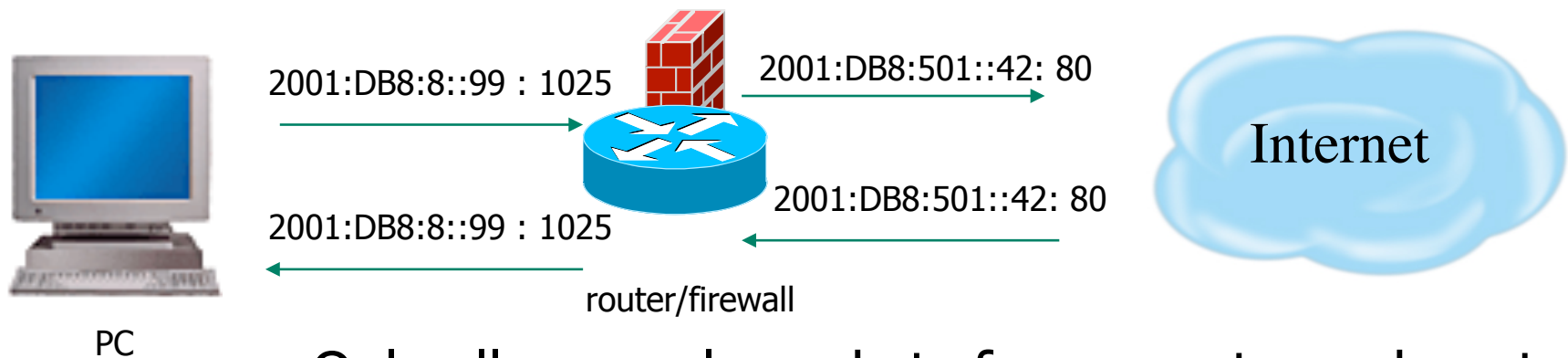


Translates `2001:DB8:501::42:40` to `2001:DB8:6::99:80`  
For IPv6, probably makes sense from IPv4 to IPv6 translation



# Stateful Inspection Firewall

- Examines the state and the context of the packets
- Remembers what outgoing requests have been sent and only allow responses to those requests back through the firewall
- Attempts to access the internal network that have not been requested by the internal network will be denied



Only allows reply packets for requests made out  
Blocks other unregistered traffic

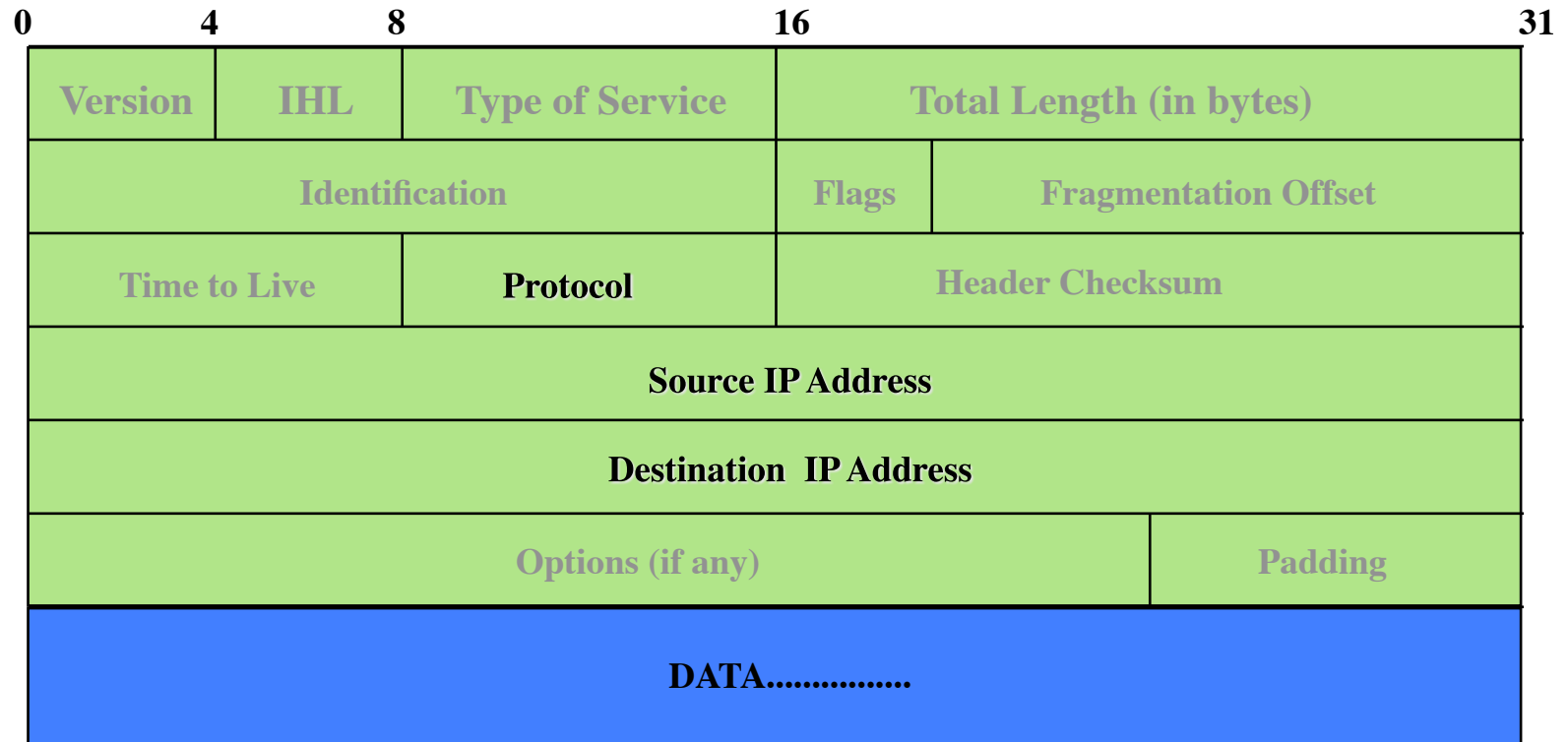


# General Filtering / Firewall BCP

- Explicitly deny all traffic and only allow what you need
- 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)



# IP Header Format



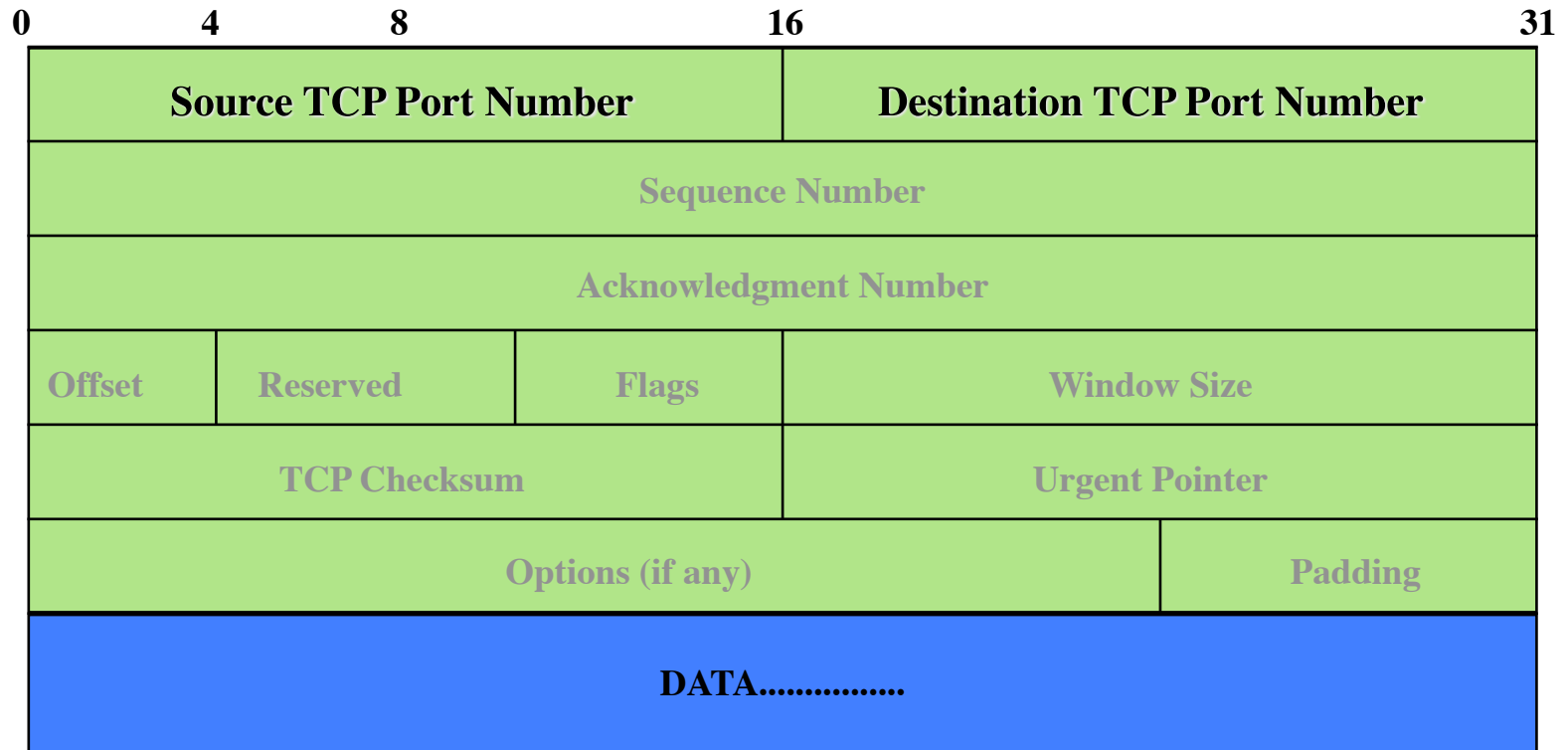


# TCP (Transport Control Protocol)

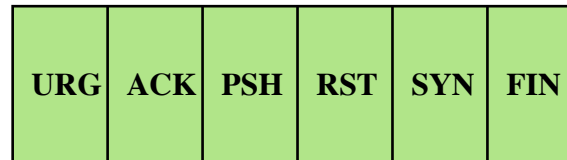
- Provides reliable virtual circuits to user processes
- Lost or damaged packets are resent
- Sequence numbers maintain ordering
- All packets except first contain ACK #  
(ACK# = sequence number of last sequential byte successfully received)
- Ports Numbers
  - Port numbers < 1024 are privileged ports
  - Destination port is fixed
  - Source port is randomly generated



# TCP Header Format



# TCP Control Flags



- URG: indicates urgent data in data stream
- ACK: acknowledgement of earlier packet
- PSH: flush packet and not queue for later delivery
- RST: reset connection due to error or other interruption
- SYN: used during session establishment to synchronize sequence numbers
- FIN: used to tear down a session

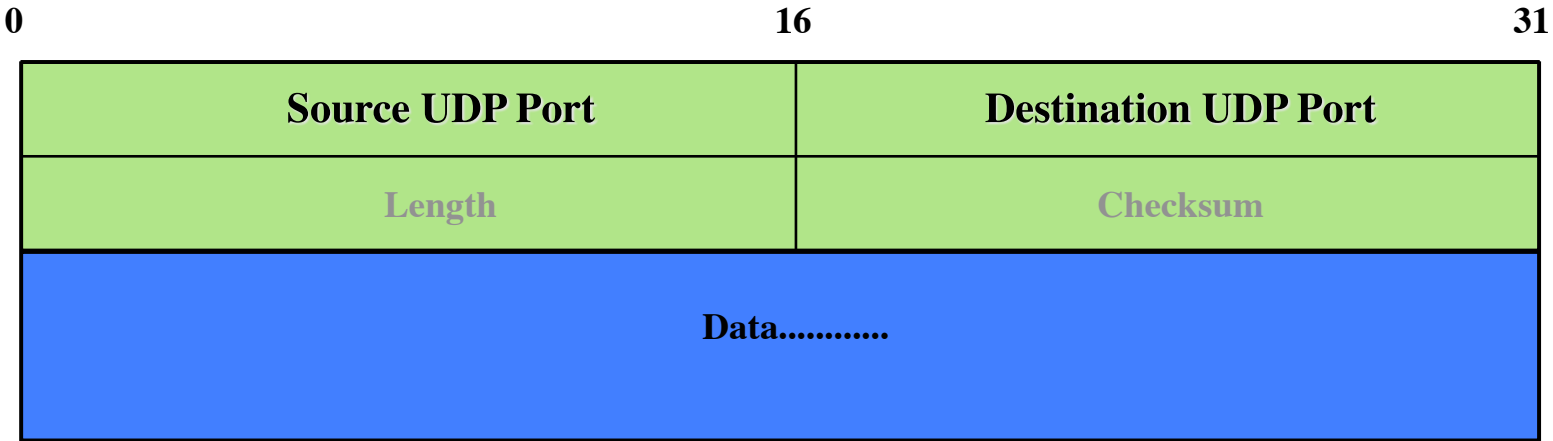


# UDP (User Datagram Protocol)

- Delivery is on a best-effort basis
  - No error correction
  - No retransmission
  - No lost, duplicate, re-ordered packet detection
- Easier to spoof than TCP packets
  - No handshake
  - No sequence numbers



# UDP Header Format



# ICMP

- Transmits command and control information
  - ICMP Echo
    - determines whether another system is alive
  - ICMP Destination Unreachable
    - No route to destination
  - ICMP Source Quench
    - Slow down number of packets sent



# ICMP

- IP Hdr and first 64 bits of transport header
  - included in ICMP Message
  - limits scope of changes dictated by ICMP
  - older implementations do not use this info
    - Destination Unreachable messages can affect all connections between a pair of hosts
    - Redirect messages should only be obeyed by hosts (from router or directly connected network)



# ICMP Message Types

Message Type	Value	Description
<b>Echo Reply</b>	<b>0</b>	Ping response if system alive
<b>Destination Unreachable</b>	<b>3</b>	Earlier IP message not deliverable
Source Quench	4	Packets received too fast to process
Redirect	5	Traffic should be directed to another router
<b>Echo</b>	<b>8</b>	Send a ping
<b>Time Exceeded</b>	<b>11</b>	Max # of hops in TTL field is exceeded
Parameter Problem	12	Bad parameter in header field
Timestamp	13	Includes time on sending machine and requests time on destination machine
Timestamp Reply	14	Timestamp response
Information Request	15	Used by host to determine which network it is on
Information Reply	16	Contains response to information request





# IP Fragment Issues

- Only first fragmented packet contains port number information
- Firewall should have capability of fragment reassembly



# Fragmentation Overlap

- Description
  - reassembly algorithms result in new fragments overwriting any overlapped portions of previously-received fragments
- Exploit
  - an attacker could construct a series of packets in which the lowest (zero-offset) fragment would contain innocuous data (and thereby be passed by packet filters)
  - If some subsequent packet has a non-zero offset it would overlap TCP header information and cause it to be modified
  - The second packet would be passed through most filter implementations because it does not have a zero fragment offset.



# Fragmentation Overlap Example

1. The filter is configured to drop TCP connection request packets.
2. The first fragment contains values, e.g., SYN=0, ACK=1, that enable it to pass through the filter unharmed.
3. The second fragment, with a fragment offset of eight octets, contains TCP Flags that differ from those given in the first fragment, e.g., SYN=1, ACK=0.
4. Since this second fragment is not a 0-offset fragment, it will not be checked, and it, too will pass through the filter.
5. The receiving host, if it conforms fully to the algorithms given in [RFC 791](#), will reconstitute the packet as a connection request because the "bad" data arrived later.



# Tiny Fragments

- Description
  - It is possible to impose an unusually small fragment size on outgoing packets.
- Exploit
  - If the fragment size is made small enough to force some of a TCP packet's TCP header fields into the second fragment, filter rules that specify patterns for those fields will not match.
  - If the filtering implementation does not enforce a minimum fragment size, a disallowed packet might be passed because it didn't hit a match in the filter

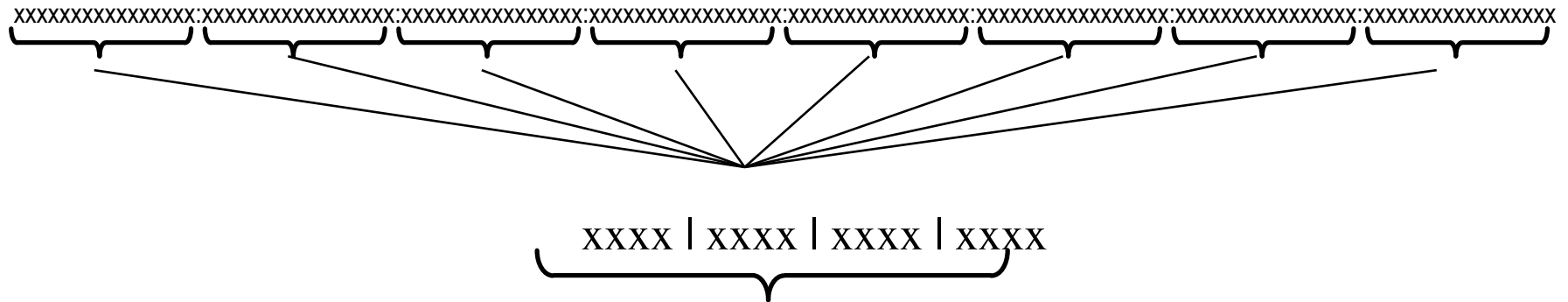


# Tiny Fragment Example

- The first fragment contains only eight octets of data (the minimum fragment size).
- In the case of TCP, this is sufficient to contain the source and destination port numbers, but it will force the TCP flags field into the second fragment.
- Filters that attempt to drop connection requests (TCP datagrams with  $SYN=1$  and  $ACK=0$ ) will be unable to test these flags in the first octet, and will typically ignore them in subsequent fragments.



# IPv6 Addressing



0000: 0	0100: 4	1000: 8	1100: C
0001: 1	0101: 5	1001: 9	1101: D
0010: 2	0110: 6	1010: A	1110: E
0011: 3	0111: 7	1011: B	1111: F

2001:DB8::/32

2001:DB8:0:0::/64

2001:DB8:0:0:0:0:FFFE::/112



# IPv6 Addressing Architecture

- Unicast Addresses
  - Global Unique
  - Link-Local
  - Site-Local (deprecated but still used)
  - Special-Use
    - Unspecified / Loopback / IPv4-Compatible / IPv4-Mapped
- Multicast Addresses
- Interface Identifiers



# Stateless Address Autoconfiguration (SLAAC)

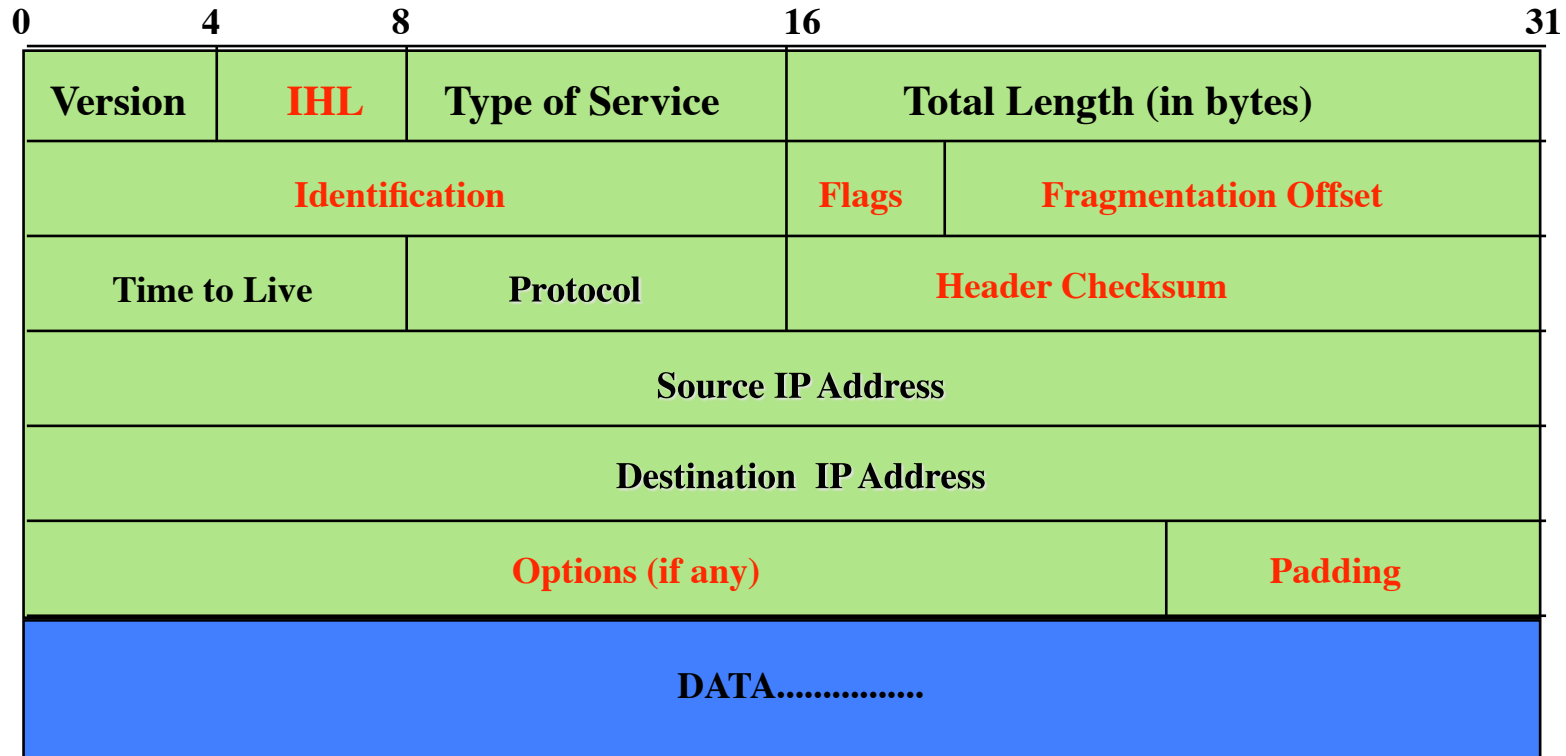
- RFC2462
- For autoconfiguration of IPv6 there are two options
  - Stateful (DHCPv6)
  - Stateless (via RA)
- For SLAAC this is done by combining address prefix advertised in the RA with the Interface ID
  - EUI-64 or RFC3041 (privacy addresses)
- Thought to help renumbering of a network
- Problem
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





# IPv4 Header Format

20 octets + options: 13 fields, including 3 flag bits

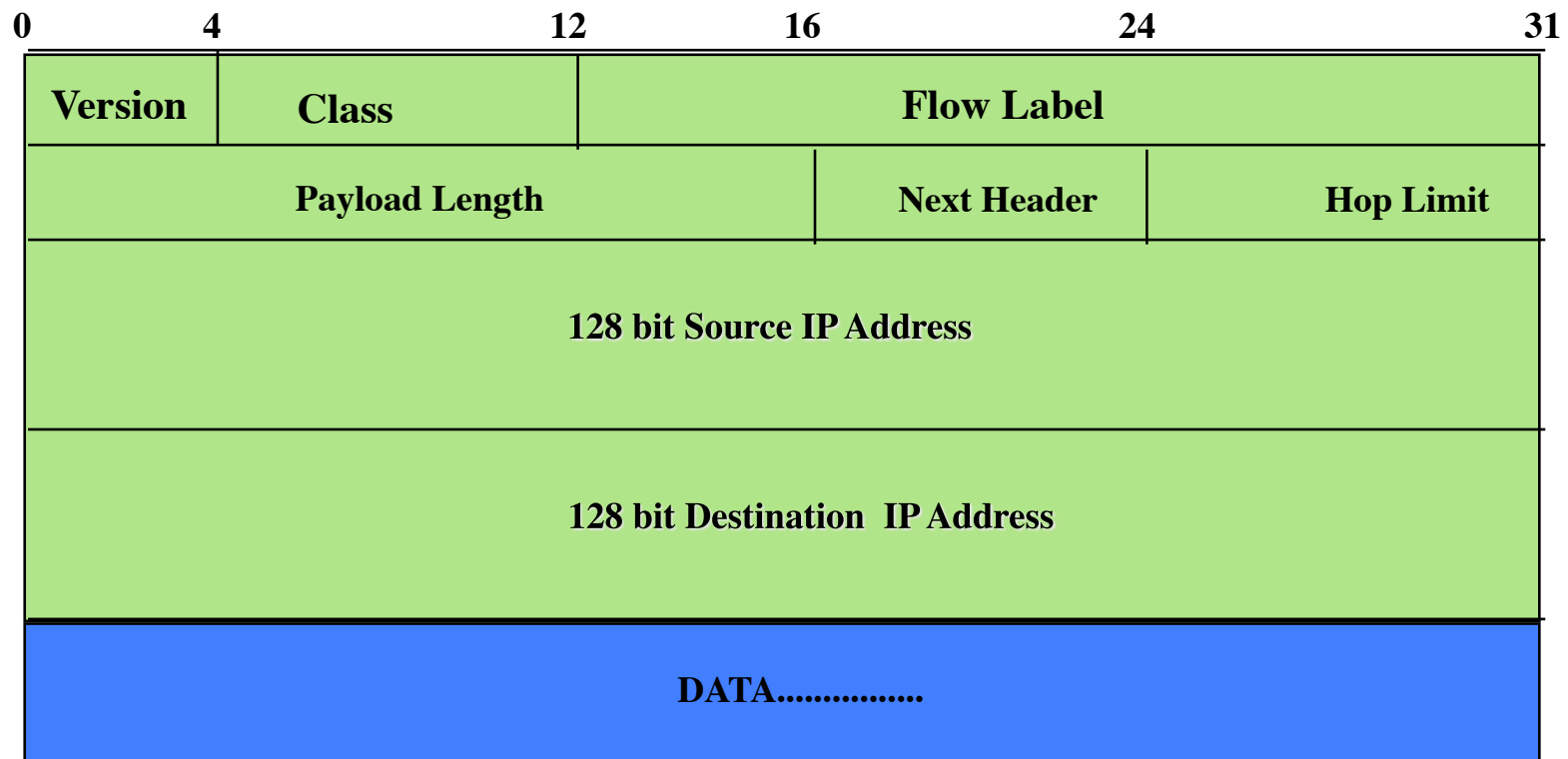


Fields in red front are absent from IPv6 header



# IPv6 Header Format

40 octets, 8 fields



# Summary: IPv4/IPv6 Header Changes

- Streamlined
  - Fragmentation fields moved out of base header
  - IP options moved out of base header
  - Header Checksum eliminated
  - Header Length field eliminated
  - Length field excludes IPv6 header
- Revised
  - Time to Live = Hop Limit
  - Protocol = Next Header
  - Precedence & TOS = Traffic Class
  - Addresses increased from 32 bits to 128 bits
- Extended
  - Flow Label field added



# IPv6 Extension Headers

- 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)

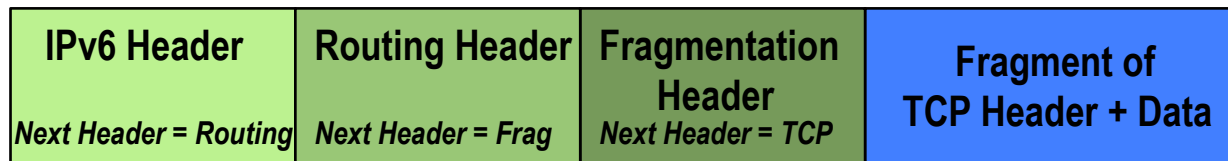
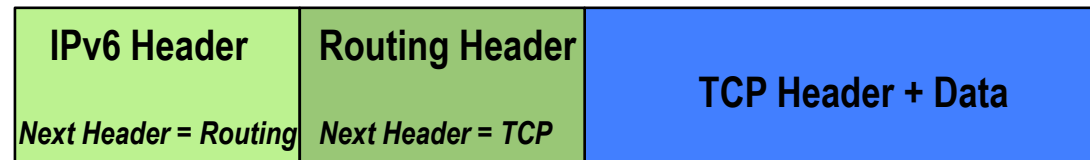
## Other Extension Header Values:

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



# Extension Header Chaining

There can be zero, one, or more Extension Headers between the IPv6 header and the upper-layer protocol header



# Extension Header Ordering

- Hop-by-Hop
- Destination Options\*
  - For options processed by the 1<sup>st</sup> destination address plus subsequent destinations listed in the routing header
- Routing
- Fragment
- Authentication
- Encapsulating Security Payload
- Destination Options
- Upper-Layer header



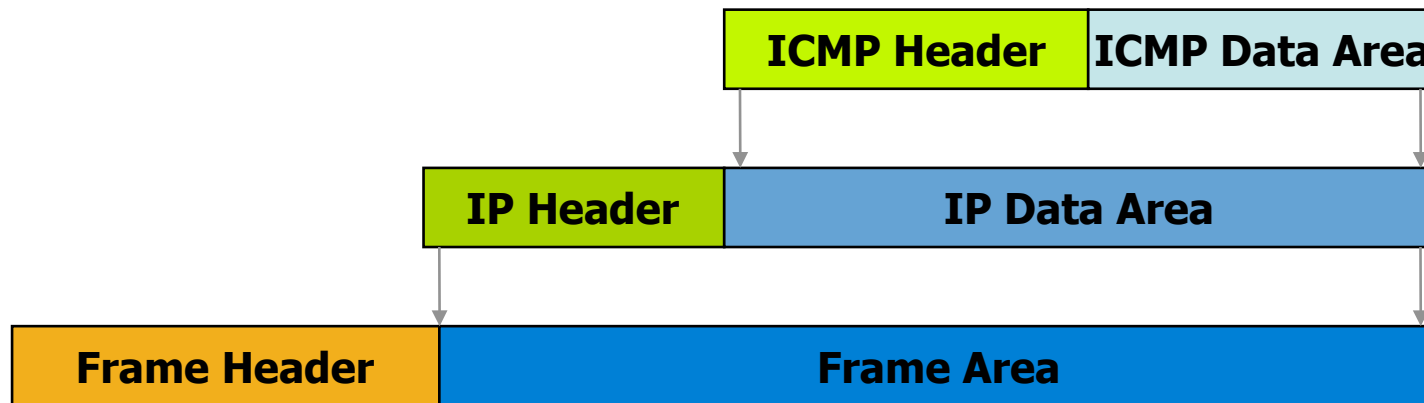
# IPv6 Fundamentals Review (ICMP Considerations)



Security Workshop – MENOG 4 Manama, Bahrain - April 2009

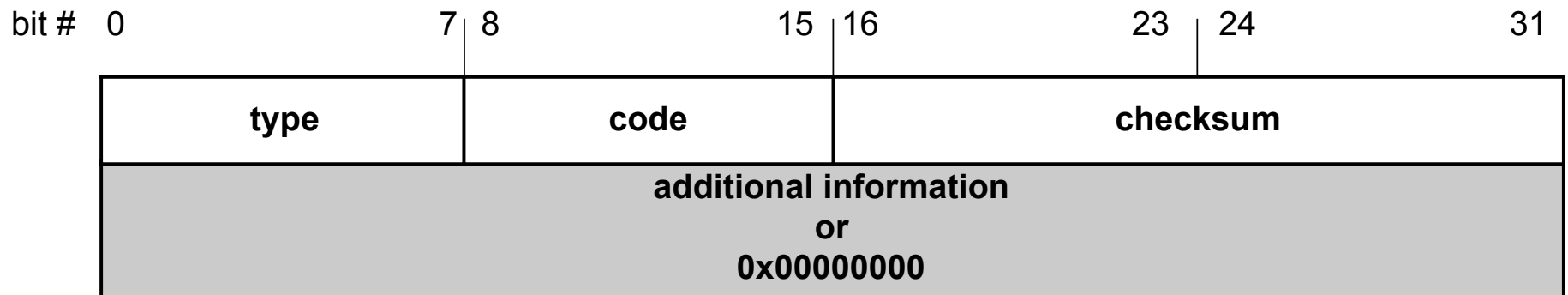
# Internet Control Message Protocol

- Original specification in RFC 792
- Used to report problems with delivery of IP packets
- Supports Path MTU (PMTU) Discovery between a sender and a receiver, which helps to optimize performance of data delivery between pairs or hosts by avoiding fragmentation en route





# ICMP Message Format



## 4 byte header:

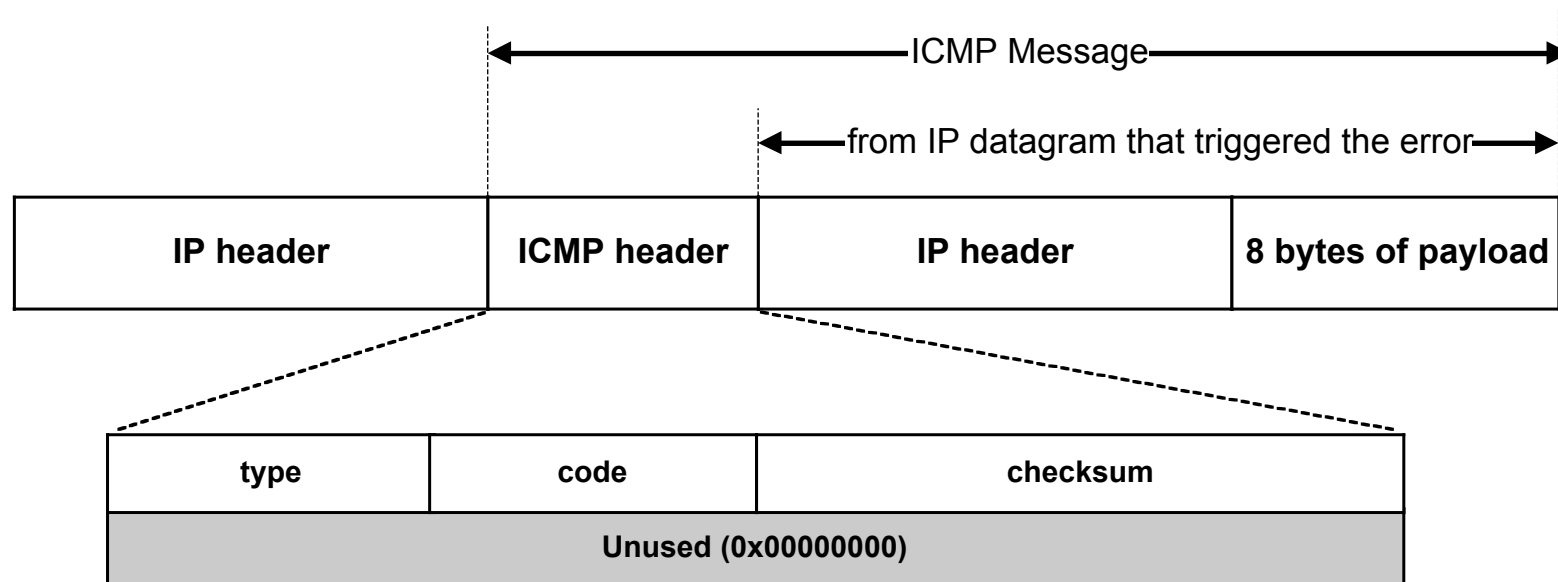
- **Type (1 byte):** type of ICMP message
- **Code (1 byte):** subtype of ICMP message
- **Checksum (2 bytes):** similar to IP header checksum. Checksum is calculated over entire ICMP message

If there is no additional data, there are 4 bytes set to zero.  
Each ICMP messages is at least 8 bytes long



# ICMP Error Messages

- ICMP error messages report error conditions
- Typically sent when a datagram is discarded
- ICMP error messages include the complete IP header and the first 8 bytes of the payload (typically: UDP, TCP)



# ICMP Message Types

Type	Message Type	Description
3	Destination Unreachable	Packet could not be delivered
11	Time Exceeded	Time to live field hit 0
12	Parameter Problem	Invalid header field
4	Source Quench	Tell host to slow down transmission due to congestion
5	Redirect	Notification that packet seems to be routed wrong
8	Echo	Ask a machine if it is alive and reachable
0	Echo Reply	Yes, I am alive
13	Timestamp Request	Same as Echo request, but with timestamp
14	Timestamp Reply	Same as Echo reply, but with timestamp



# Destination Unreachable Codes

Code	Definition
0	Network Unreachable [no routing table entry available for destination network]
1	Host Unreachable [destination host should be directly reachable but does not respond to ARP requests]
2	Protocol Unreachable [protocol in protocol field of IP header is not supported at destination]
3	Port Unreachable [transport protocol at destination host cannot pass datagram to an application]
4	Fragmentation needed & Don't Fragment was set
5	Source Route failed
6	Destination Network Unknown
7	Destination Host Unknown
8	Source Host Isolated
9	Communication Destination Network is Administratively Prohibited
10	Communication Destination Host is Administratively Prohibited
11	Destination Network Unreachable for Type of Service
12	Destination Host Unreachable for Type of Service
13	Communication Administratively Prohibited
14	Host Precedence Violation
15	Precedence Cutoff Violation



# Additional Codes for ICMP Types

## Redirect Codes

Code	Definition
0	Redirect Datagram for the Network (or subnet)
1	Redirect Datagram for the Host
2	Redirect Datagram for the Type of Service & Network
3	Redirect Datagram for the Type of Service & Host

## Time Exceeded Codes

Code	Definition
0	Time to Live Exceeded in Transit
1	Fragment Reassembly Time Exceeded

## Parameter Problem Codes

Code	Definition
0	Pointer Indicates the Error
1	Missing a Required Option
2	Bad Length



# ICMPv6

- Is similar to IPv4 ICMP, with a few differences:
  - ICMP is carried in an IPv6 datagram
  - A checksum is computed since ICMP is a transport protocol, relative to IPv6
  - New messages are defined for the IPv6 specification
  - In an error message, the original datagram is included in the error packet for easier recovery by the source
- Identified by the Next Header value = 58
- ICMP header contains Type and Code fields to identify and qualify the message specifics
- Two defined ICMP classes in IPv6:
  - Error Messages
  - Informational



# ICMPv6 Error Messages

- Identified by a Type field value between 0 and 127
- Message Types:
  - destination unreachable
    - no route
    - administratively prohibited (i.e. firewalls)
    - address unreachable
    - port unreachable
  - packet too big
  - time exceeded
  - parameter problem
    - erroneous header field
    - unrecognized next header type
    - unrecognized option



# ICMPv6 Informational Messages

## (Type: 128-255)

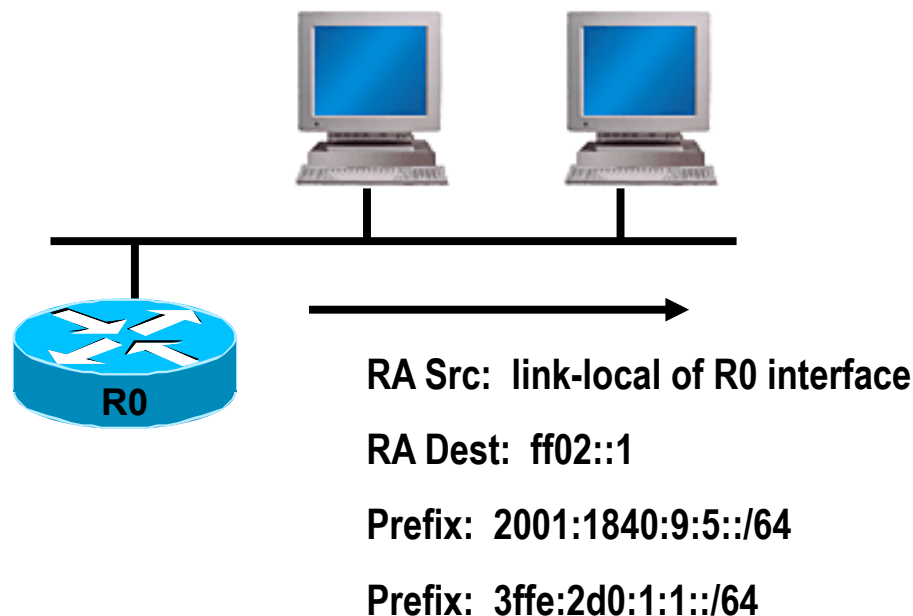
<i>ICMP Number</i>	<i>Message Type</i>
128	Echo request
129	Echo reply
130	Multicast group membership query
131	Multicast group membership report
132	Multicast group membership termination
133	Router Solicitation
134	Router Advertisement
135	Neighbor Solicitation
136	Neighbor Advertisement
137	Redirect
138	Router Renumbering
139	Node Information query
140	Node Information reply
141	Inverse Neighbor Solicitation
142	Inverse Neighbor Advertisement





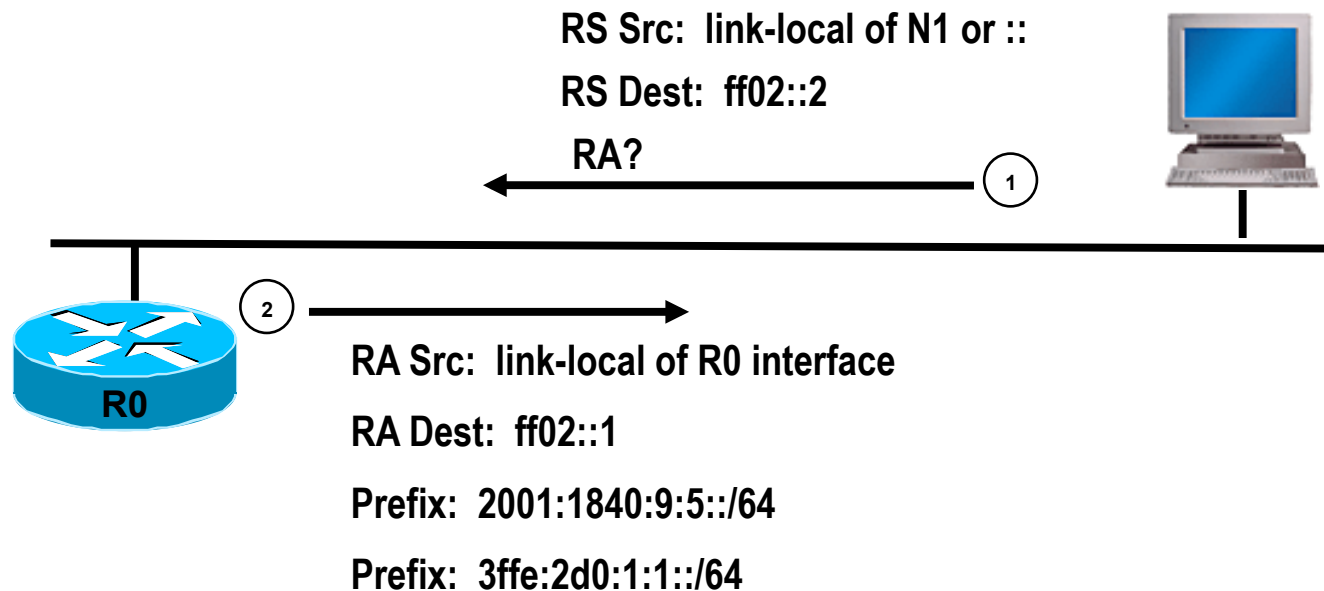
# IPv6 Router Advertisement

- 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



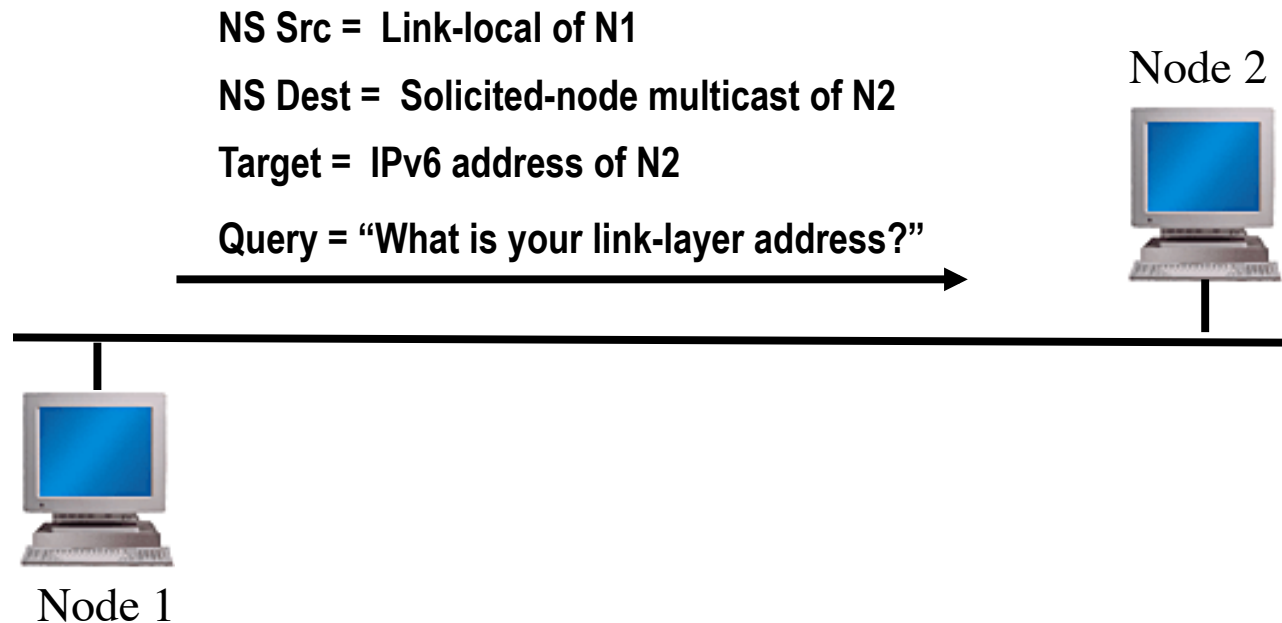
# IPv6 Router Solicitation

- 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



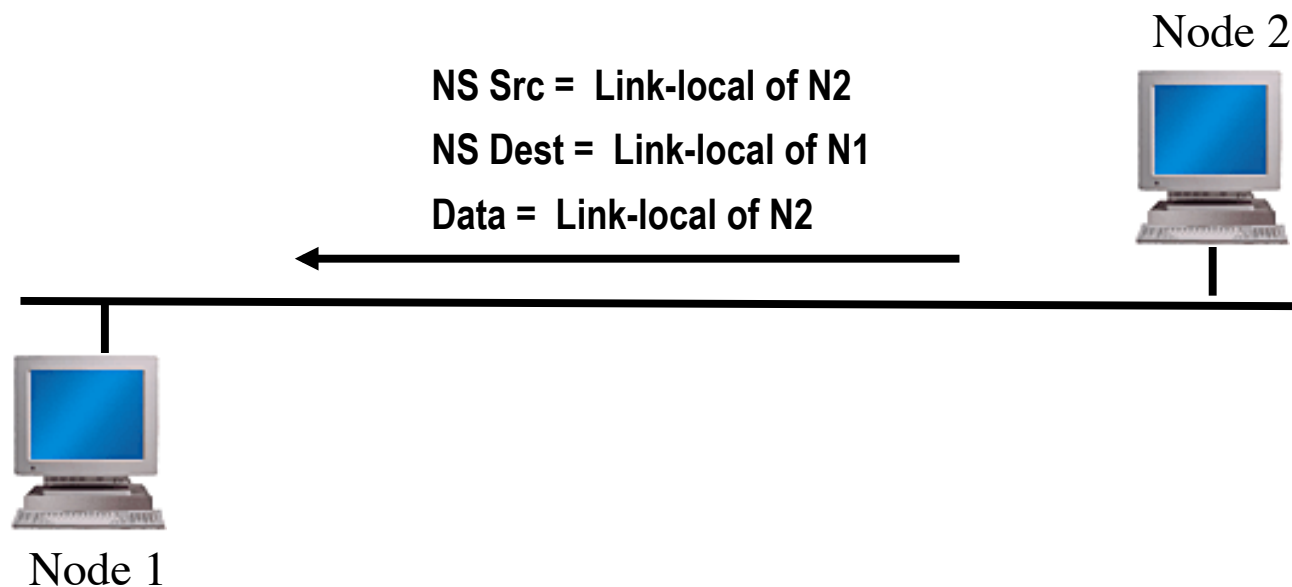
# IPv6 Neighbor Solicitation

- 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



# 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.



# Filtering Recommendations

- Block incoming loopback packets and RFC 1918 networks
  - 127.0.0.0
  - 10.0.0.0 – 10.255.255.255
  - 172.16.0.0 – 172.31.0.0
  - 192.168.0.0 – 192.168.255.255
- Block multicast packets (if NOT using multicast)
- Block broadcast packets (careful of DHCP & BOOTP users)
- Block incoming packets that claim to have same destination and source address



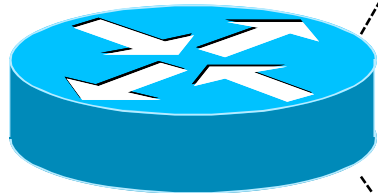
# DoS Filtering

(\* these networks were reallocated and are actually used)

Description	Network
default	0.0.0.0 /8
loopback	127.0.0.0 /8
RFC 1918	10.0.0.0 /8
RFC 1918	172.16.0.0 /12
RFC 1918	192.168.0.0 /16
Net Test	192.0.2.0 /24
Testing devices *	192.18.0.0 /15
IPv6 to IPv4 relay *	192.88.99.0 /24
RFC 1918 nameservers *	192.175.48.0 /24
End-node auto configuration *	169.254.0.0 /16



# Example Incoming IPv4 Bogon Packet Filter



```
ip access-list extended DSL-Incoming
deny ip 127.0.0.0 0.255.255.255 any log
deny ip 0.0.0.0 0.255.255.255 any log
deny ip 10.0.0.0 0.255.255.255 any log
deny ip 169.254.0.0 0.0.255.255 any log
deny ip 172.16.0.0 0.15.255.255 any log
deny ip 192.168.0.0 0.0.255.255 any log
deny ip 224.0.0.0 15.255.255.255 any log
permit icmp any any ttl-exceeded
permit icmp any any echo-reply
permit icmp any any echo
permit tcp any any eq 22 log
permit udp host <ip address> eq domain <subnet range>
permit udp host <ip address> eq domain <subnet range>
permit udp host <ip address> <subnet range> eq ntp
permit udp host <ip address> <subnet range> eq ntp
deny tcp any any eq 443
deny tcp any any eq 139
deny tcp any any eq 445
deny tcp any any eq 2967
permit tcp any <my sybnet> established
deny ip any any log
```





# RFC2827 (BCP38) – Ingress Filtering

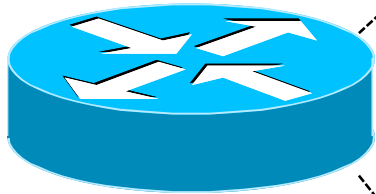
If an ISP is aggregating routing announcements for multiple downstream networks, strict traffic filtering should be used to prohibit traffic which claims to have originated from outside of these aggregated announcements.

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.



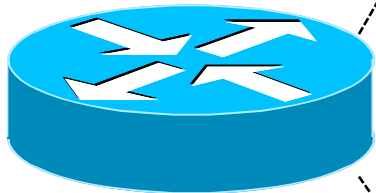
# Example Outgoing Packet Filter



```
ip access-list extended DSL-Outbound
permit tcp host <my host ip address> eq ftp-data any log
permit tcp host <my host ip address> eq ftp any log
permit ip <my subnet> any
deny ip any any log
```



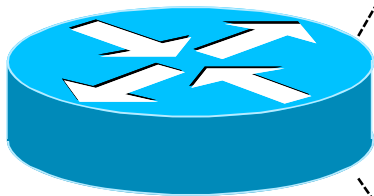
# Example Edge Filter (part 1)



```
access-list 100 permit icmp any any echo-reply
access-list 100 permit icmp any any echo
access-list 100 permit icmp any any ttl-exceeded
access-list 100 permit icmp any any unreachable
access-list 100 deny icmp any any
access-list 100 deny tcp any any eq www
access-list 100 deny tcp any any eq 1080
access-list 100 deny tcp any any eq 3127
access-list 100 deny tcp any any eq 3128
access-list 100 deny tcp any any eq 1433
access-list 100 deny tcp any any eq 4662
access-list 100 deny tcp any any eq 6881
access-list 100 deny udp any any eq 6881
access-list 100 deny udp any eq 6881 any
access-list 100 deny tcp any eq 6881 any
access-list 100 deny tcp any any eq 8080
access-list 100 deny tcp any any eq 20157
access-list 100 deny tcp any any eq 38402
access-list 100 permit tcp any any established
access-list 100 permit tcp any any eq 22
access-list 100 permit tcp any any eq bgp
access-list 100 permit udp any any eq domain
```



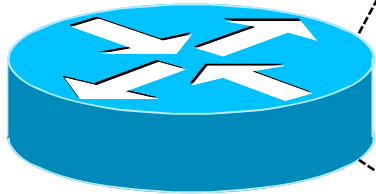
# Example Edge Filter (part 2)



```
access-list 100 permit tcp any any eq ident
access-list 100 permit udp any any eq ntp
access-list 100 permit udp any eq ntp any
access-list 100 permit udp any any eq 5
access-list 100 permit udp any eq isakmp any
access-list 100 deny  udp any any eq 2049
access-list 100 permit udp any any gt 1023
access-list 100 permit ipinip any any
access-list 100 permit 41 any any
access-list 100 permit esp any any
access-list 100 permit gre any any
access-list 100 deny  ip any any log
!
access-list 101 permit icmp any any echo-reply
access-list 101 permit icmp any any echo
access-list 101 permit icmp any any ttl-exceeded
access-list 101 permit icmp any any unreachable
access-list 101 deny  icmp any any
access-list 101 deny  udp any any eq netbios-ns
access-list 101 deny  tcp any any eq 135
access-list 101 deny  tcp any any eq 139
access-list 101 deny  udp any any eq netbios-dgm
```



# Example Edge Filter (part 3)



```
access-list 101 deny  udp any eq 6881 any
access-list 101 deny  udp any any eq 6881
access-list 101 permit ipinip any any
access-list 101 permit 41 any any
access-list 101 permit esp any any
access-list 101 permit gre any any
access-list 101 permit ip any any
access-list compiled
```

!

```
Interface FastEthernet0/0
description Link to ISP
ip access-group 100 in
ip access-group 101 out
```



# 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



# General Firewall BCP

- Explicitly deny all traffic and only allow what you need
- 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 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
- 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 Packet Filters To Consider(cont.)

- 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 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 (`2002::/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.



# Allow Following ICMPv6 Through A 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

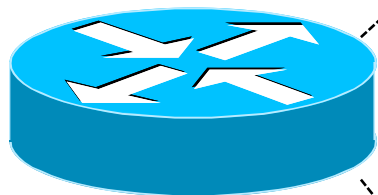


# Allow Following ICMPv6 To/From A Firewall

- 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



# Example Outgoing Packet Filter



```
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
```



# Cisco Filters (Access-Lists)

- IPv6 access-lists (ACL) are used to filter traffic and restrict access to the router
- IPv6 extended access lists add support for option header and upper layer filtering
- Cisco specific filtering characteristics
  - A reference to an empty ACL will permit any any
  - Implicit permit rule for neighbor discovery
  - Implicit deny any any as final rule in each ACL



# Configuring Cisco IPv6 ACLs

- Creating the IPv6 ACL

```
[no] ipv6 access-list <name>
```

- Defining the IPv6 ACL entry

```
[no] permit | deny ipv6 | <protocol> any | host <src> | src/len [sport] any | host <dest> |  
dest/len [dport] [reflect <name> [timeout <secs>]] [fragments] [routing] [dscp <val>]  
[flow-label <val>] [time-range <name>] [log | log-input] [sequence <num>]
```

- Applying an ACL to an interface

```
interface s0/0  
ipv6 traffic-filter ipv6_in in  
ipv6 traffic-filter ipv6_out out
```

- Restricting access to the router

```
line vty 0 4  
ipv6 access-class vty-filter in
```



# Monitoring Cisco IPv6 ACLs

- Show the IPv6 ACL configuration

```
show ipv6 access-list [name]
```

- Clearing the IPv6 ACL match count

```
clear ipv6 access-list [name]
```

*As with any filter configuration, ordering is important since all entries are checked sequentially. Ensure that most frequent 'hits' are on top of the list.*





# Routing Header: RFC 2460 Text

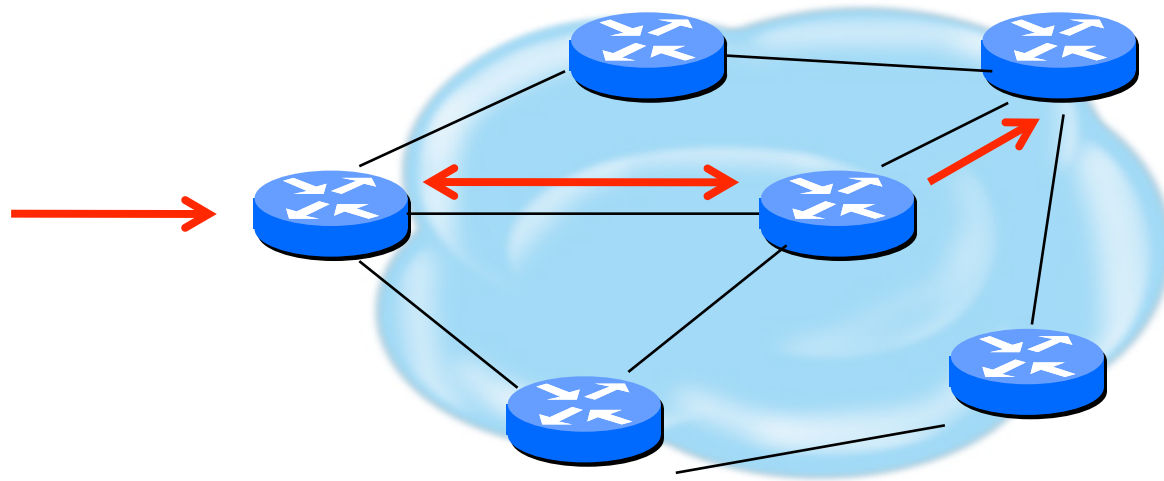
- The routing header is used by an IPv6 source to list one or more intermediate **nodes** to be “visited” on the way to packet’s destination.
- Each extension header should occur at most once, except for the destination options header which should occur at most twice.
- IPv6 nodes must accept and attempt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



Security Workshop – MENOG 4 Manama, Bahrain - April 2009

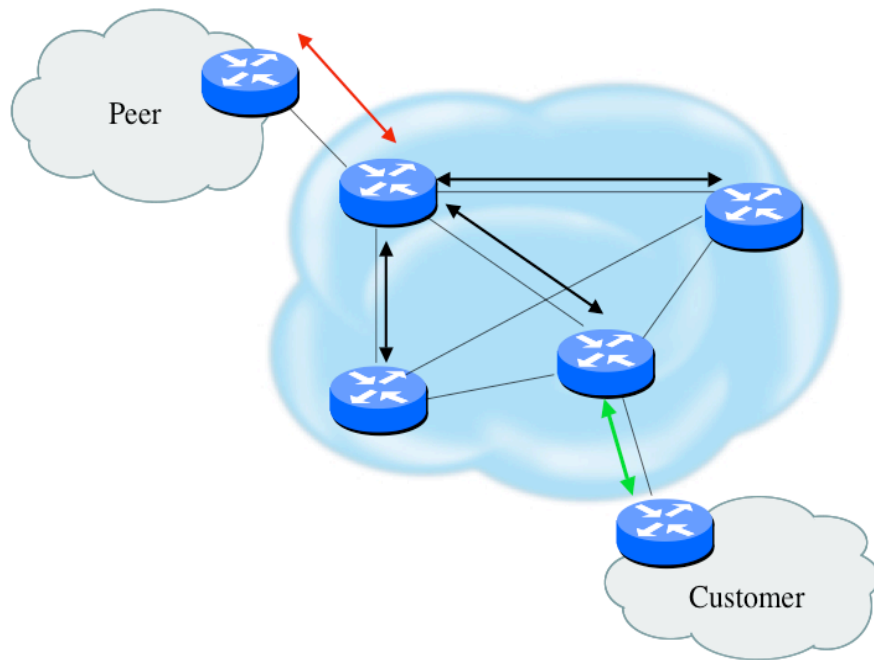
# Router Security Considerations

- 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)

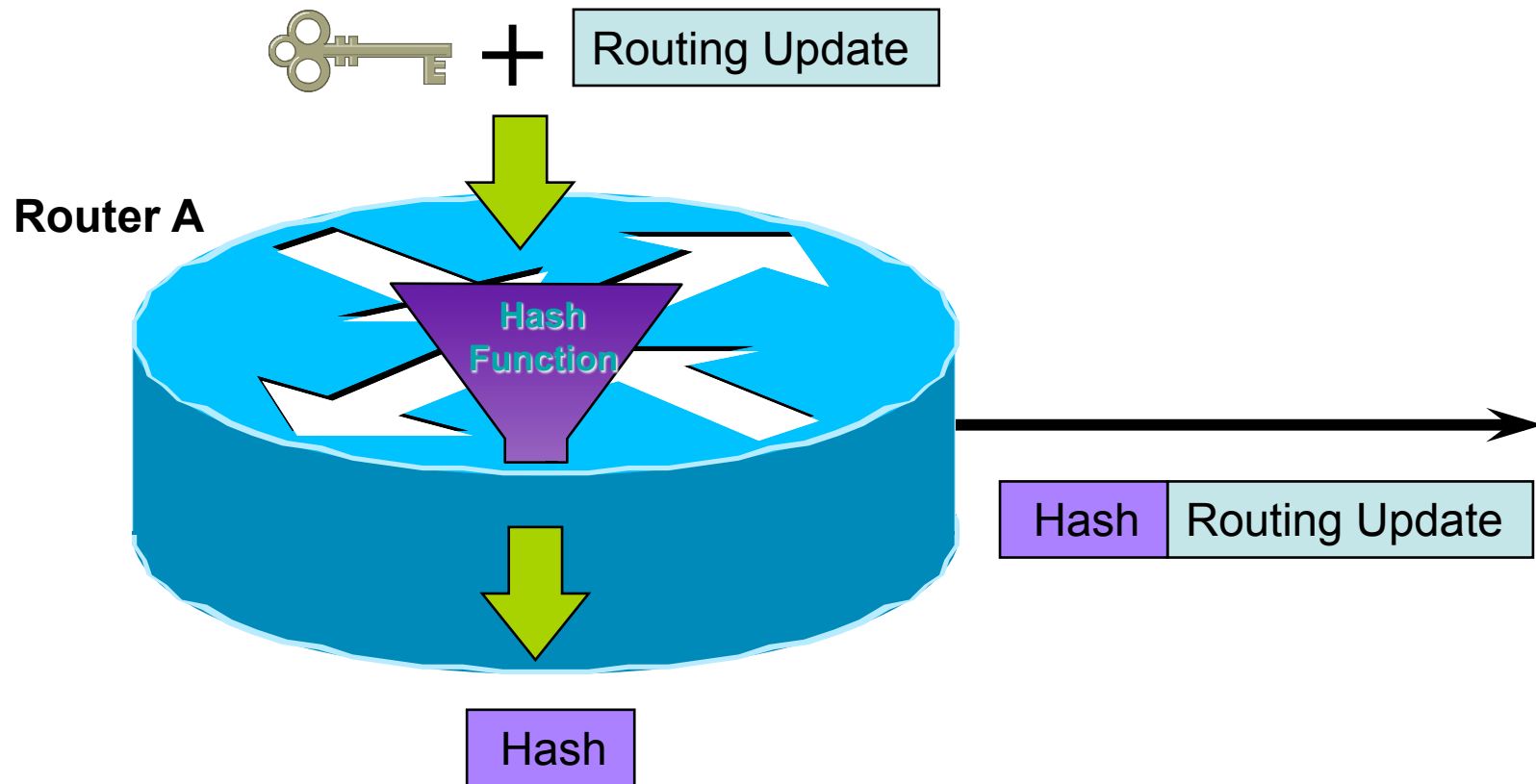


# Basics of Hash Algorithms

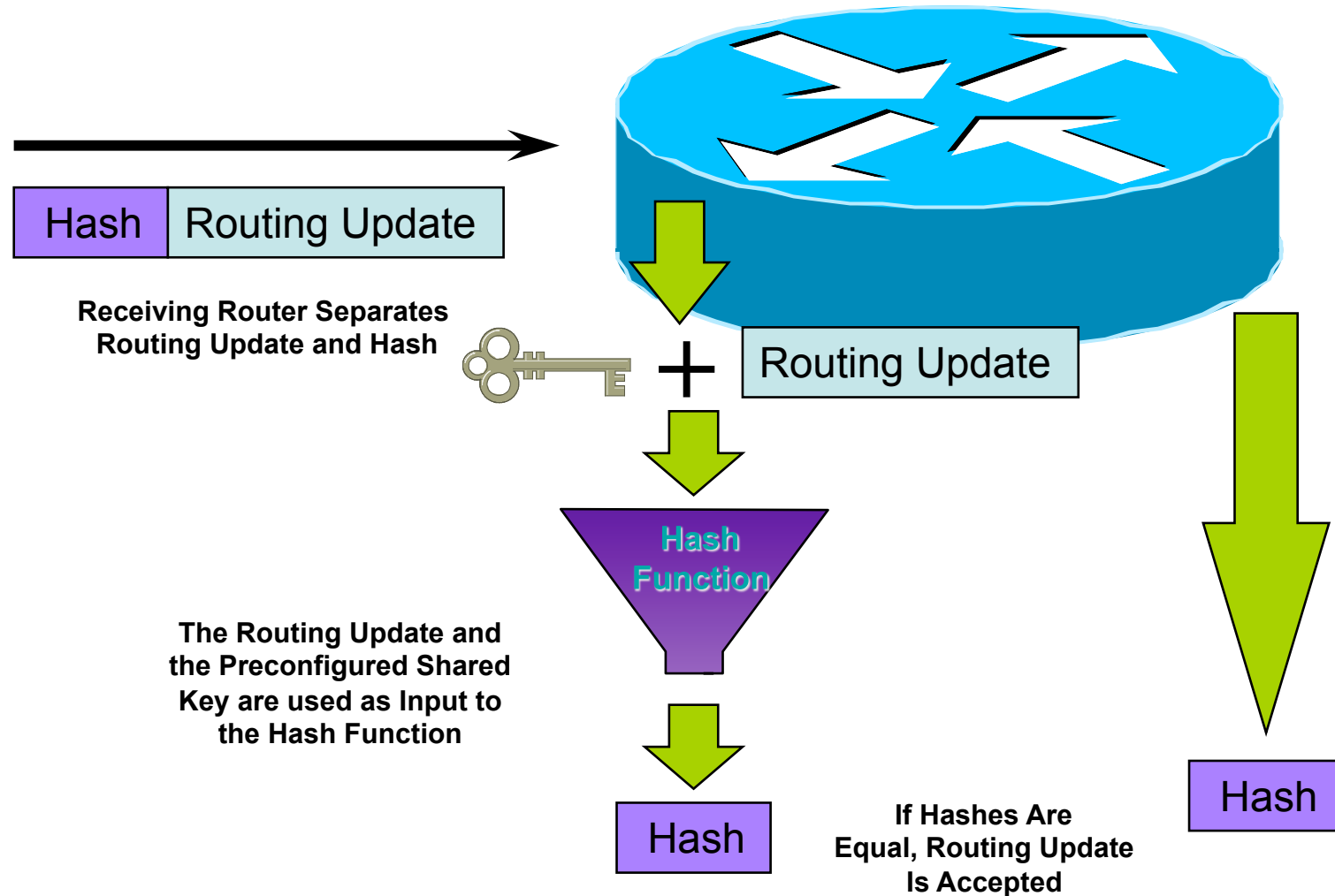
- Reduces a variable-length input to a fixed-length output
  - Output is called a *hash* or *message digest* or *fingerprint*
  - Output length is 128 bits for MD5 and 160 bits for SHA-1
- Requirements
  - Can't deduce input from output
  - Can't generate a given output
  - Can't find two inputs which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



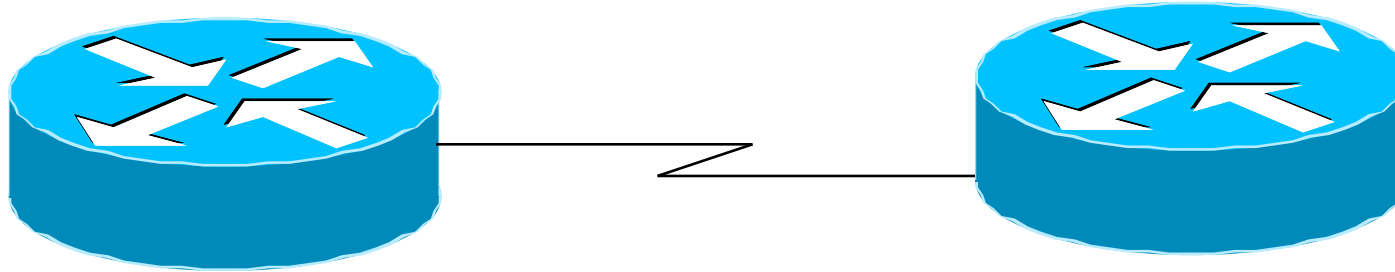
# MD-5 Based Authentication



# MD-5 Based Authentication



# Sample MD-5 Auth Configuration (OSPF)



```
interface Loopback0  
ip address 70.70.70.70 255.255.255.255
```

```
interface Serial2  
ip address 192.16.64.2 255.255.255.0
```

```
ip ospf message-digest-key 1 md5 mk6  
router ospf 10  
network 192.16.64.0 0.0.0.255 area 0  
network 70.0.0.0 0.255.255.255 area 0  
area 0 authentication message-digest
```

```
interface Loopback0  
ip address 172.16.10.36 255.255.255.240
```

```
interface Serial1/0  
ip address 192.16.64.1 255.255.255.0
```

```
ip ospf message-digest-key 1 md5 mk6  
router ospf 10  
network 172.16.0.0 0.0.255.255 area 0  
network 192.16.64.0 0.0.0.255 area 0  
area 0 authentication message-digest
```



# 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



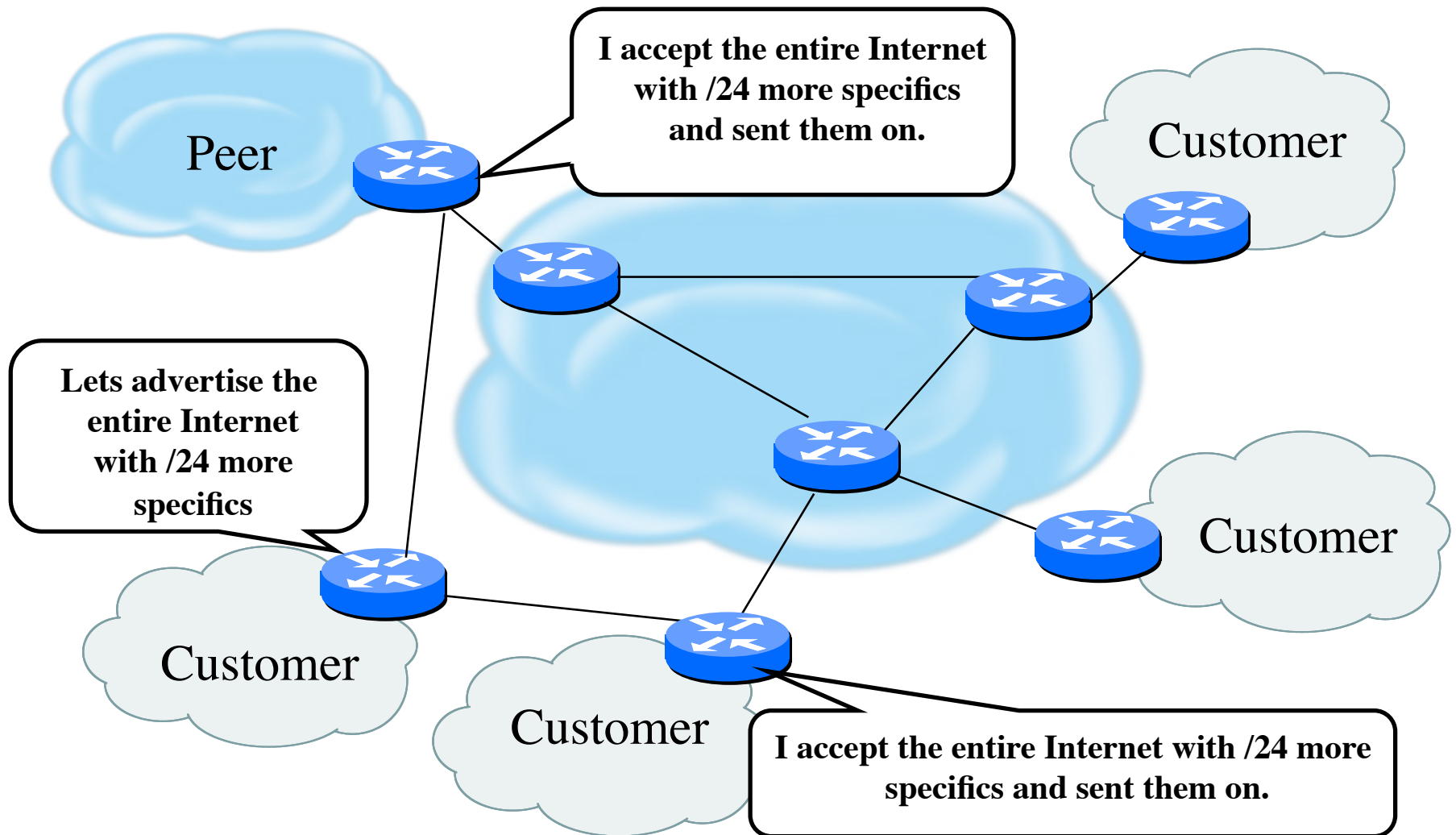


# BGP Prefix Filtering

- 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.

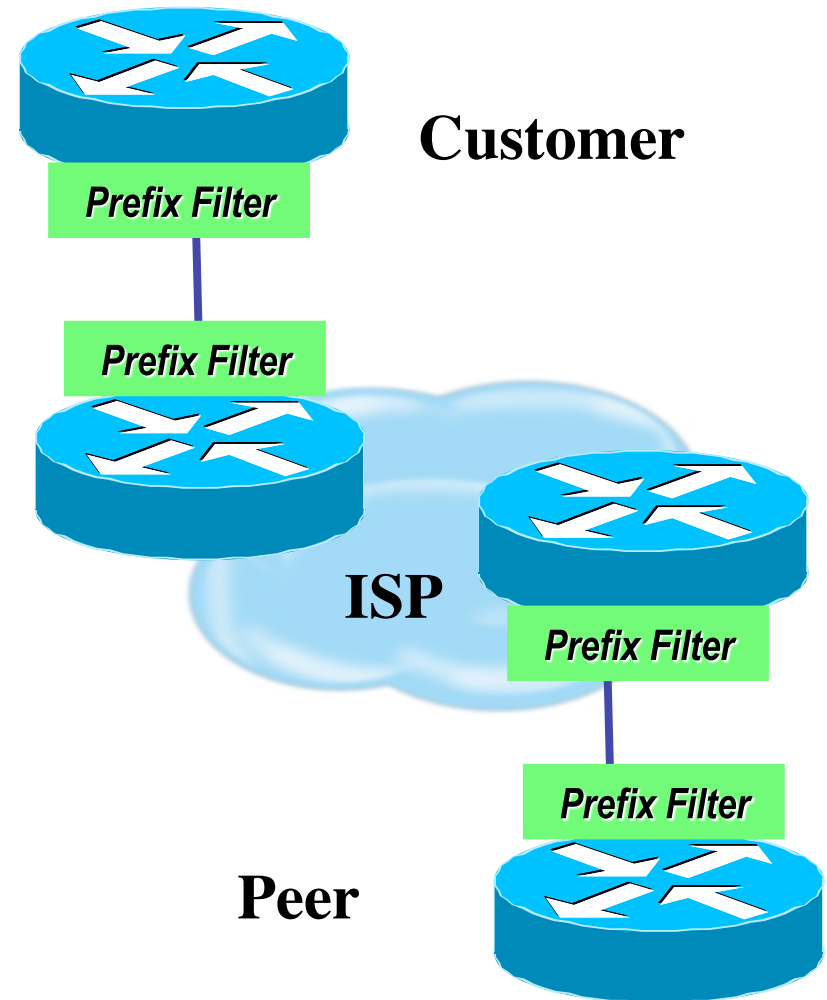


# Example: No Prefix Filtering

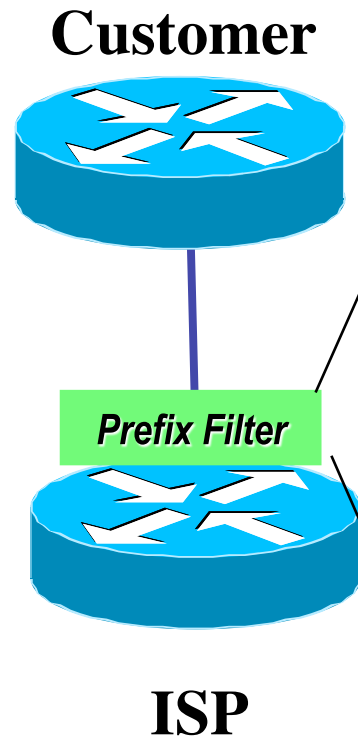


# Where to Prefix Filter ?

- Customer's Ingress/Egress
- ISP Ingress on Customer (may Egress to Customer)
- ISP Egress to Peer and Ingress from Peer
- Peer Ingress from ISP and Egress to ISP



# Receiving Customer Prefixes



```
router bgp 100
  neighbor 123.123.6.1 remote-as 101
  neighbor 123.123.6.1 prefix-list customer in
  !
  ip prefix-list customer permit 121.60.0.0/2
  ip prefix-list customer deny 0.0.0.0/0 le 32
```



# Peering With Other ISPs

- Similar to eBGP customer aggregation except inbound prefix filtering is rarely used (lack of global registry)
- Use maximum-prefix and prefix sanity checking instead
- Still use per-neighbor passwords!



# Example of ISP-Peers (peer group)

*neighbor nap peer-group*

*neighbor nap description for peer ISPs*

*neighbor nap remove-private-AS*

*neighbor nap version 4*

*neighbor nap prefix-list sanity-check in*

*neighbor nap prefix-list cidr-block out*

*neighbor nap route-map nap-out out*

*neighbor nap maximum prefix 30000*



# Example of ISP Peers route-map

```
route-map nap-out permit 10  
match community 1 ; customers only  
set metric-type internal ; MED = IGP metric  
set ip next-hop peer-address ; our own
```



# Sanity Check Prefix List (part 1)

# FIRST - FILTER OUT YOUR IGP ADDRESS SPACE!!

# deny the default route

ip prefix-list sanity-check seq 5 deny 0.0.0.0/32

# deny anything beginning with 0

ip prefix-list sanity-check seq 10 deny 0.0.0.0/8 le 32

# deny masks > 20 for all class A nets (1-127)

ip prefix-list sanity-check seq 15 deny 0.0.0.0/1 ge 20

# deny 10/8 per RFC1918

ip prefix-list sanity-check seq 20 deny 10.0.0.0/8 le 32





# Sanity Check Prefix List (part 2)

# reserved by IANA - loopback address

ip prefix-list sanity-check seq 25 deny 127.0.0.0/8 le 32

# deny masks  $\geq 17$  for all class B nets (129-191)

ip prefix-list sanity-check seq 30 deny 128.0.0.0/2 ge 17

# deny net 128.0 - reserved by IANA

ip prefix-list sanity-check seq 35 deny 128.0.0.0/16 le 32

# deny 172.16 as RFC1918

ip prefix-list sanity-check seq 40 deny 172.16.0.0/12 le 32

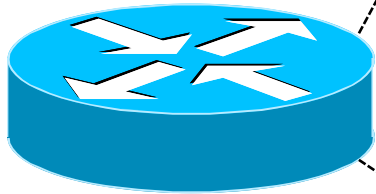


# Sanity Check Prefix List (part 3)

```
# class C 192.0.20.0 reserved by IANA
ip prefix-list sanity-check seq 45 deny 192.0.2.0/24 le 32
# class C 192.0.0.0 reserved by IANA
ip prefix-list sanity-check seq 50 deny 192.0.0.0/24 le 32
# deny 192.168/16 per RFC1918
ip prefix-list sanity-check seq 55 deny 192.168.0.0/16 le 32
# deny 191.255.0.0 - IANA reserved
ip prefix-list sanity-check seq 60 deny 191.255.0.0/16 le 32
# deny masks > 25 for class C (192-222)
ip prefix-list sanity-check seq 65 deny 192.0.0.0/3 ge 25
# deny anything in net 223 - IANA reserved
ip prefix-list sanity-check seq 70 deny 223.255.255.0/24 le 32
# deny class D/Experimental
ip prefix-list sanity-check seq 75 deny 224.0.0.0/3 le 32
```



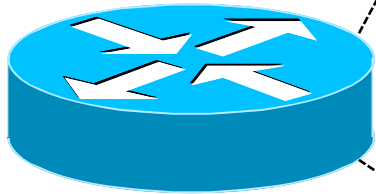
# BGP Configuration for Bogon Filtering



```
Router bgp 300
No synchronization
  neighbor x.x.x.x remote-as 66
  neighbor x.x.x.x version 4
  neighbor x.x.x.x prefix-list bogon-filter in
  neighbor x.x.x.x prefix-list bogon-filter out
  neighbor z.z.z.z remote-as 99
  neighbor z.z.z.z version 4
  neighbor z.z.z.z prefix-list bogon-filter in
  neighbor z.z.z.z prefix-list bogon-filter out
no auto-summary
```



# Example: Bogon Filter



```
ip prefix-list bogon-filter deny 0.0.0.0/8 le 32
ip prefix-list bogon-filter deny 10.0.0.0/8 le 32
ip prefix-list bogon-filter deny 127.0.0.0/8 le 32
ip prefix-list bogon-filter deny 169.254.0.0/16 le 32
ip prefix-list bogon-filter deny 172.16.0.0/12 le 32
ip prefix-list bogon-filter deny 192.0.2.0/24 le 32
ip prefix-list bogon-filter deny 192.168.0.0/16 le 32
ip prefix-list bogon-filter deny 224.0.0.0/3 le 32
ip prefix-list bogon-filter deny 0.0.0.0/0 le 32
```



# 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>

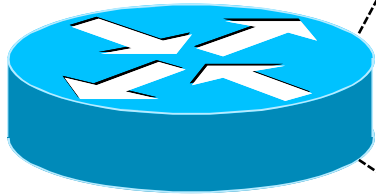


# IPv6 BGP Prefix Filters To Consider

- 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::/32` or 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



# BGP Simple 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
```



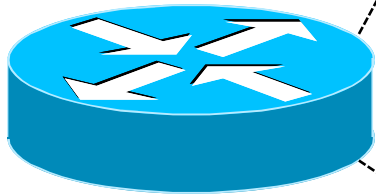
# BGP Prefix Filters (RIR Allocations)

- APNIC
  - <ftp://ftp.apnic.net/stats/apnic/delegated-apnic-latest>
- RIPE NCC
  - <ftp://ftp.ripe.net/pub/stats/ripenncc/delegated-ripenncc-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>





# BGP 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
```



# Other BGP Security Techniques

- BGP Community Filtering
- MD5 Keys on the eBGP and iBGP Peers
- Max Prefix Limits
- Prefer Customer Routes over Peer Routes (RFC 1998)
- GTSM (i.e. TTL Hack)



# Audit and Validate Your Routing Infrastructures

- Are appropriate paths used?
  - check routing tables
  - verify configurations
- Is router compromised?
  - check access logs



# Routing Security Conclusions

- Current routing protocols do not have adequate security controls
- Mitigate risks by using a combination of techniques to limit access and authenticate data
- Be vigilant in auditing and monitoring your network infrastructure
- Consider MD5 authentication
- Always filter routing updates....especially be careful of redistribution



# Mitigating DDoS Attacks



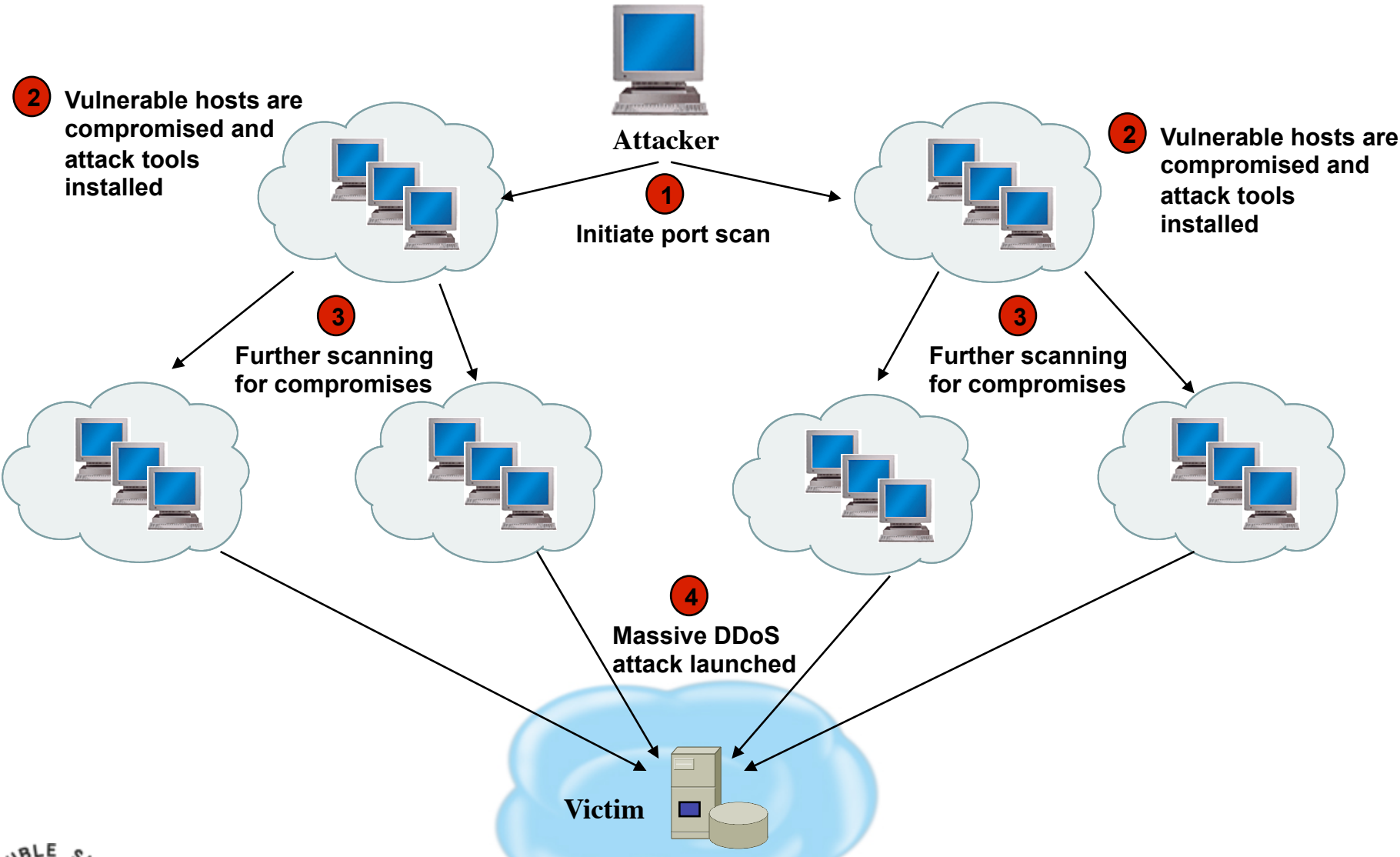
Security Workshop – MENOG 4 Manama, Bahrain - April 2009

# DDoS Is A Huge Problem

- Distributed and/or coordinated attacks
  - Increasing rate and sophistication
- Infrastructure protection
  - Coordinated attack against infrastructure
  - Attacks against multiple infrastructure components
- Overwhelming amounts of data
  - Huge effort required to analyze
  - Lots of uninteresting events



# Automated Distributed Denial of Service Attack



# Router CPU Vulnerabilities

## CPU Overload

- Attacks on applications on the Internet have affected router CPU performance leading to some BGP instability
- 100,000+ hosts infected with most hosts attacking routers with forged-source packets
- Small packet processing is taxing on many routers ...even high-end
- Filtering useful but has CPU hit





# DoS Filtering

(\* these networks may be reallocated)

Description	Network
default	0.0.0.0 /8
loopback	127.0.0.0 /8
RFC 1918	10.0.0.0 /8
RFC 1918	172.16.0.0 /12
RFC 1918	192.168.0.0 /16
Net Test	192.0.2.0 /24
Testing devices *	192.18.0.0 /15
IPv6 to IPv4 relay *	192.88.99.0 /24
RFC 1918 nameservers *	192.175.48.0 /24
End-node auto configuration *	169.254.0.0 /16



# Today's DoS Prevention

- Allow only good traffic into your network (ingress filtering)
- Allow only good traffic out of your network (egress filtering)
- Stop directed broadcast traffic (to avoid being an amplifier)

Deny all and permit only what's needed is most effective policy



# DoS/DDoS Tools

- Vendor provided
  - Arbor TrafGen
- Open source
  - stream
  - litestorm
  - rc8.o
  - f\_\_kscript
  - slice3



# Using IP Routing as a Security Tool

- IP Routing can be used to manipulate traffic on a network to:
  - Null0 (Black Hole)
  - Shunts
  - Sink Hole
  - Analysis Devices
  - Clean up Devices
  - Rate-Limit



# Securing The Device

- Miscreants have a far easier time gaining access to devices than you think.
- Ensure that the basic security capabilities have been configured.



# Fundamental Device Protection Summary

- Secure logical access to routers with passwords and timeouts
- Never leave passwords in clear-text
- Authenticate individual users
- Restrict logical access to specified trusted hosts
- Allow remote vty access only through ssh
- Disable device access methods that are not used
- Protect SNMP if used
- Shut down unused interfaces
- Shut down unneeded services
- Ensure accurate timestamps for all logging
- Create appropriate banners
- Test device integrity on a regular basis

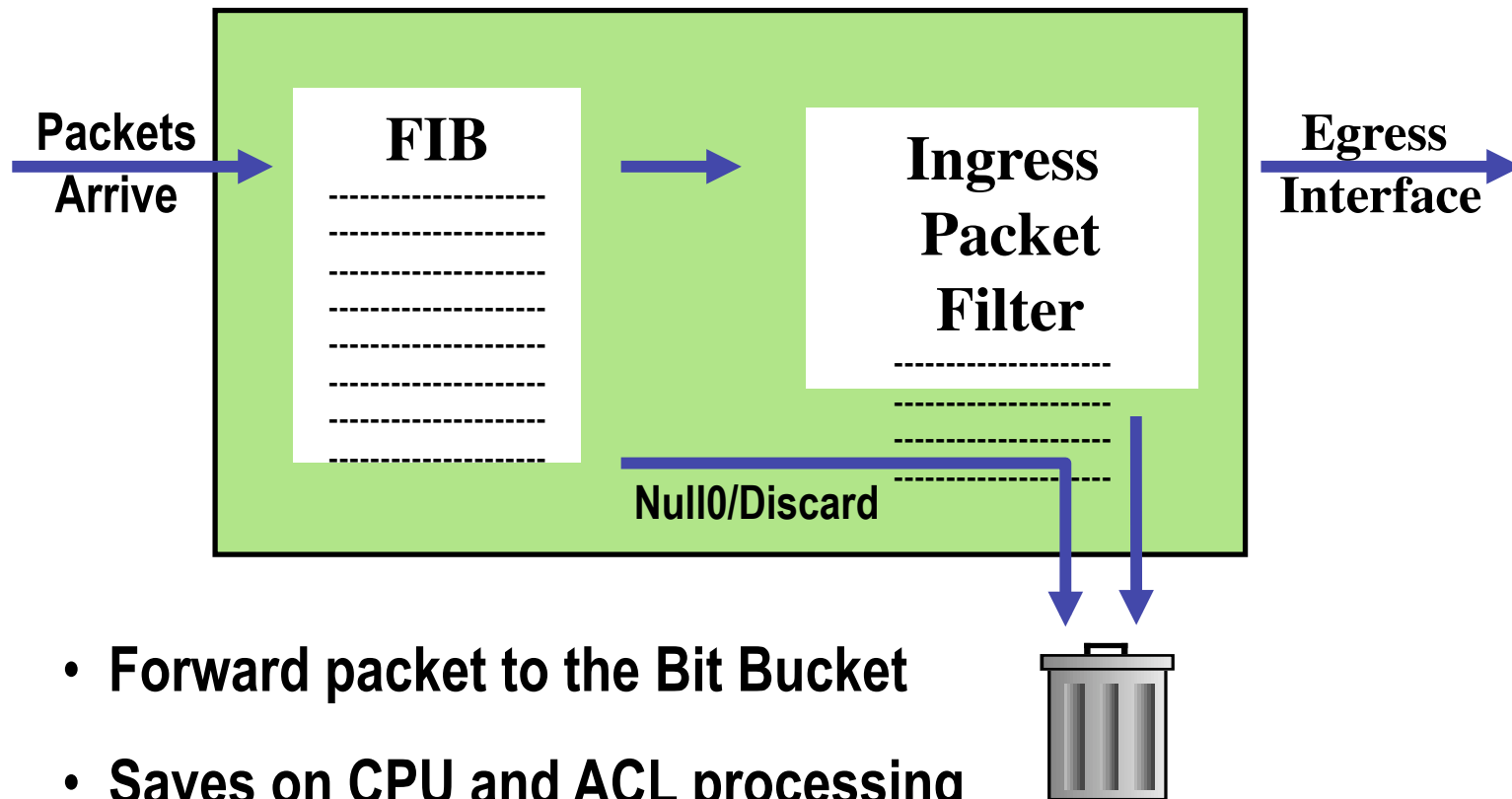


# DoS Mitigation - RTBH Basics

- Use BGP routing protocol to trigger network wide response to an attack flow.
- Simple static route and BGP allows ISP to trigger network wide black holes as fast as iBGP can update the network.
- Unicast RPF allows for the black hole to include any packet whose source or destination address matches the prefix.
- Effective against spoofed and valid source addresses.

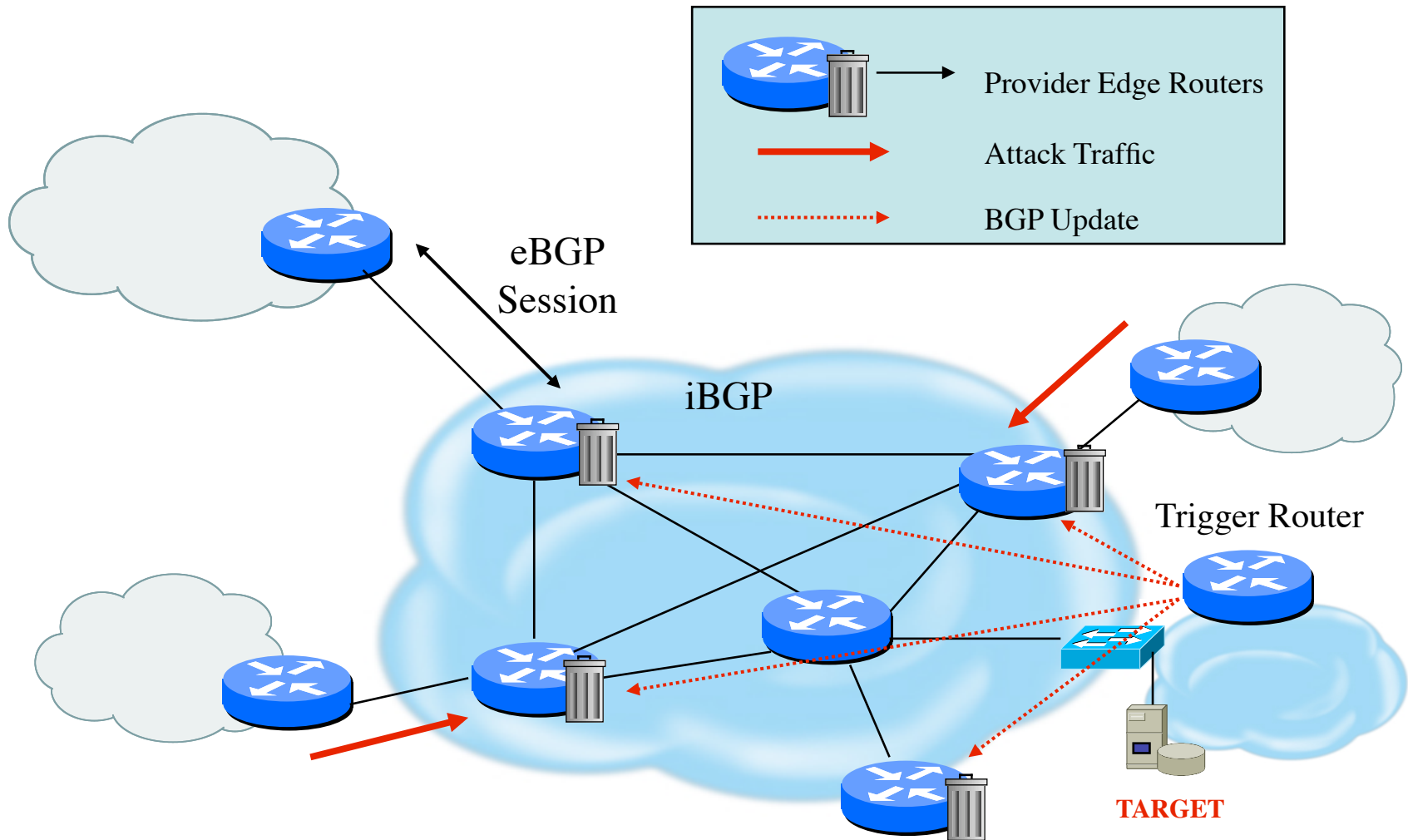


# Blackhole Filtering





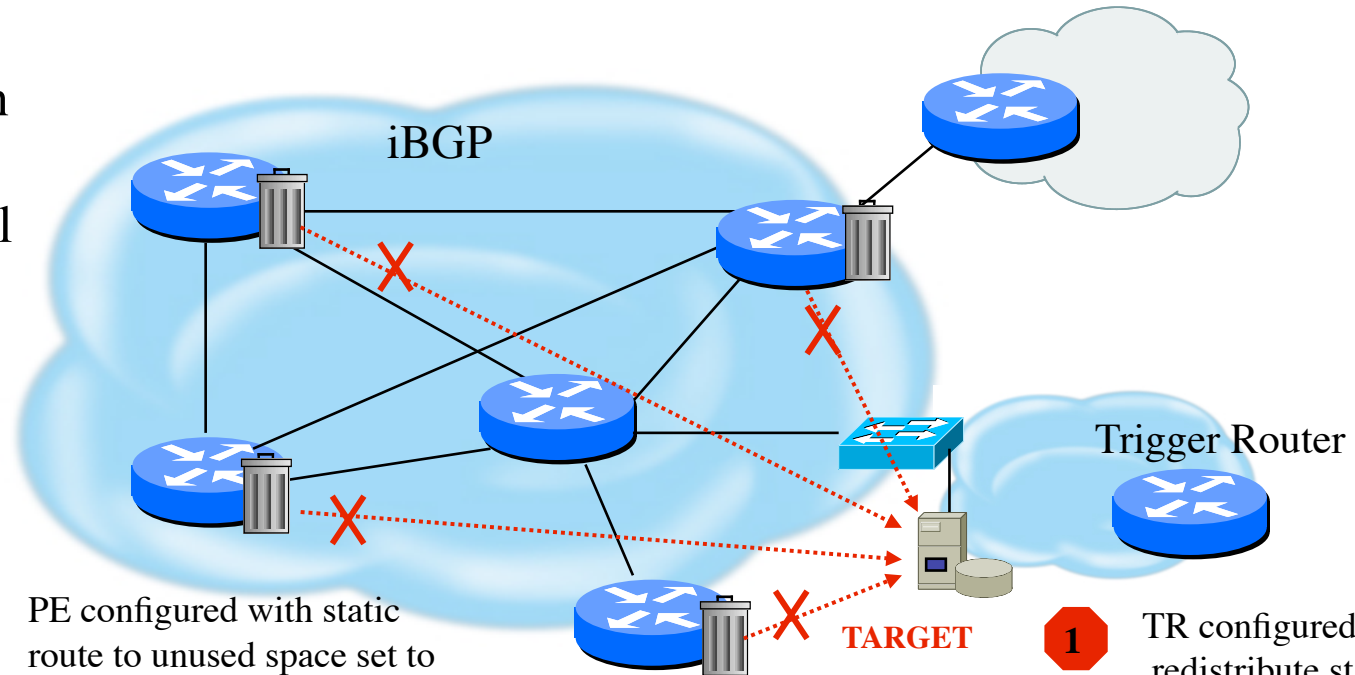
# RTBH in the Network



# Destination-Based RTBH

Steps:

1. Preparation
2. Trigger
3. Withdrawal



**1** PE configured with static route to unused space set to Null0 (192.0.2.6/32 set to Null0)

**2** Receives iBGP update which states next hop for target is 192.0.2.6/32

**3** Installs new (valid) route to target

NOTE: All traffic to the target is dropped, even legitimate traffic

**1** TR configured to redistribute static into every iBGP peer

**2** Add static route which sets next hop to target destination (192.0.2.6)

**3** Manually remove static route which causes BGP route withdrawal



# Source-Based RTBH Filtering

- Ability to drop packets at network edge based on specific source address
- Permits legitimate traffic from reaching target destination
- Depends on uRPF
- Packet dropped if:
  - If router has no entry for source IP address
  - If source IP address entry points to Null0



# uRPF Loose Mode

- Originally created to scale BCP 38 ingress filtering on the ISP
  - Customer Edge of an ISP's network.
- Loose Check Mode added to provide ISPs with means to trigger a network wide, source based black hole filter activated at BGP update speeds.
- uRPF Loose Check will passively drop any packet whose source address is not in the router's FIB (Forwarding Information Base).
- Effective way to drop Bogon addresses.



# Source Based Remote Triggered Black Hole Filtering

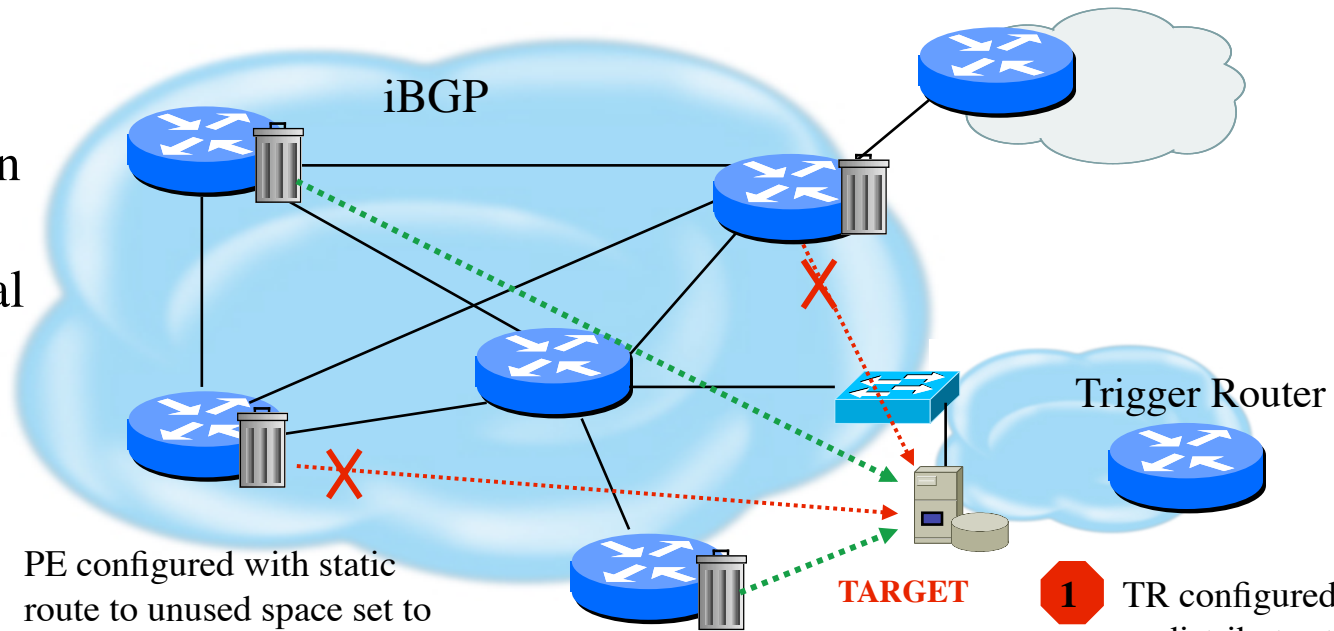
- What do we have?
  - *Black Hole Filtering* – If the destination address equals Null 0 we drop the packet.
  - *Remote Triggered* – Trigger a prefix to equal Null 0 on routers across the Network at iBGP speeds.
  - uRPF Loose Check – If the source address equals Null 0, we drop the packet.
- Put them together and we have a tool to trigger drop for any packet coming into the network whose source or destination equals Null 0!



# Source-Based RTBH

Steps:

1. Preparation
2. Trigger
3. Withdrawal



**1** PE configured with static route to unused space set to Null0 (192.0.2.6/32 set to Null0) and loose mode uRPF on external interfaces

**2** Receives iBGP update which states next hop for target is 192.0.2.6/32. All traffic from source IP will fail loose uRPF check.

**3** Installs new (valid route to target

NOTE: Only traffic from the attack sources get dropped

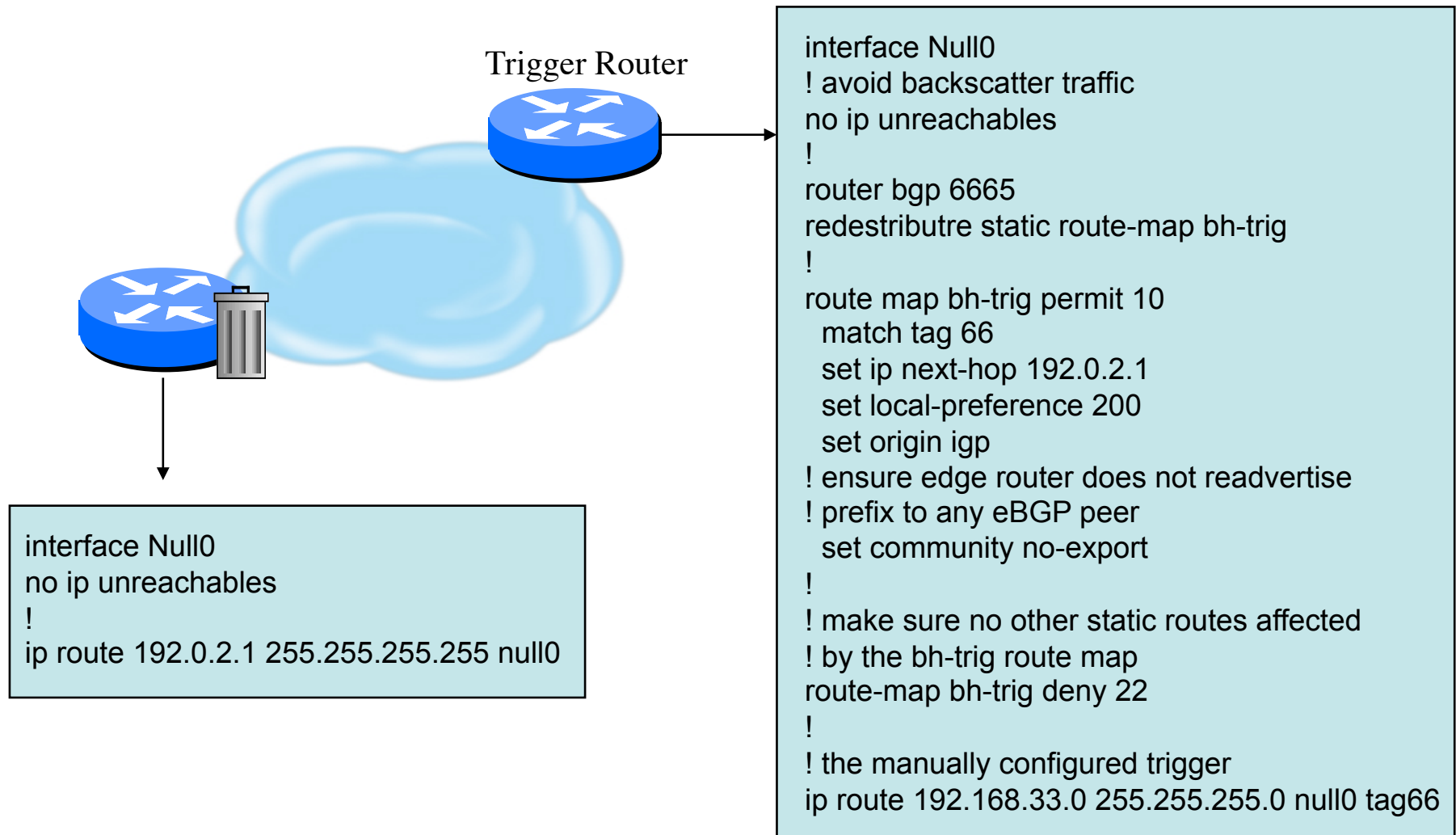
**1** TR configured to redistribute static into every iBGP peer

**2** Add static route which sets next hop to target destination (192.0.2.6)

**3** Manually remove static route which causes BGP route withdrawal



# RTBH Configuration Example



# Additional RTBH Considerations

- Avoid intentionally/unintentionally dropping legitimate traffic
- Deploy secure BGP features
  - Neighbor authentication
  - Prefix filters
  - ‘TTL hack’
- Use prefix filters at edge and trigger routers to ensure essential services (e.g. DNS) not black-holed by mistake





# Remote Triggered Drops

- Use one or both techniques to contain a worm
  - Internal deployments limit spread within enterprise
  - Edge deployments limit spread to internet and/or other external destination
- Depending on null0 location, effective quarantine tool
- Rapid reaction, highly scaleable
  - Proven technique used by large service providers



# DoS Mitigation Summary

- Consider MD-5 authentication in your routing infrastructures.
- Filter obviously bogus networks at ingress / egress points.
- Use prefix filters.
- Use remote triggered filtering techniques.
- Understand your traffic patterns and help deter attacks to downstream and upstream neighbors.



# Where Does IPsec Fit In?



Security Workshop – MENOG 4 Manama, Bahrain - April 2009

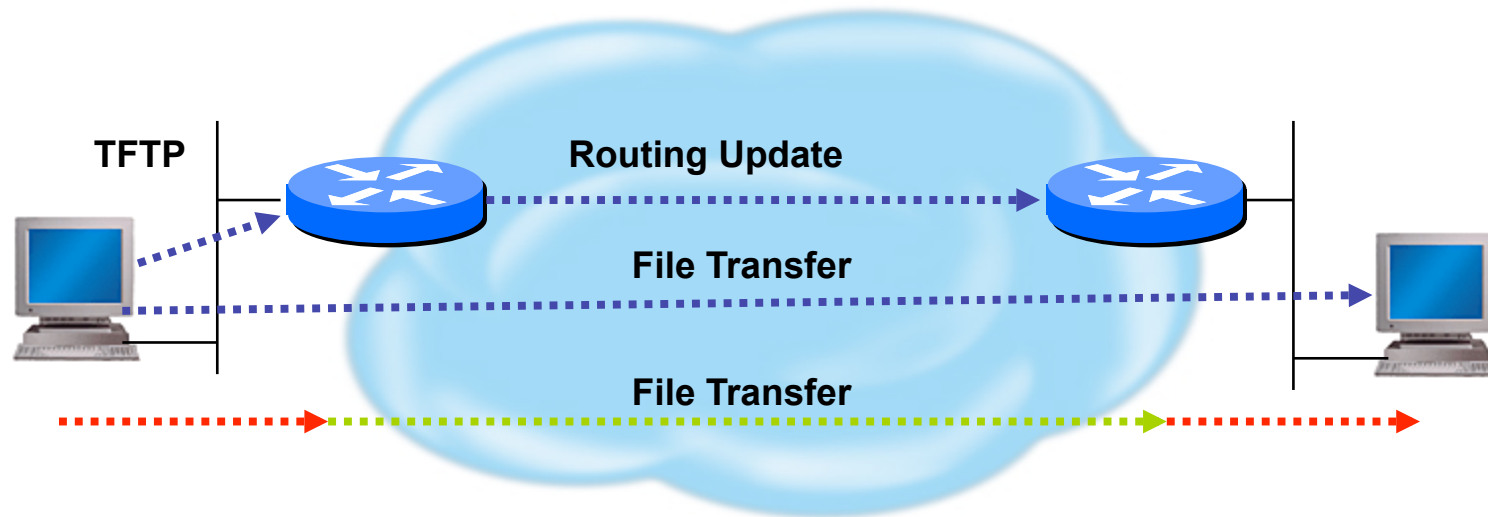
# Architectures using IPsec

- Protect all traffic using IPsec for data origin authentication and integrity
  - AH versus ESP/Null Encryption
- Add confidentiality as dictated by security policy
  - ESP

**Need to dispel myth that using IPsec mandates the demise of network layer defense mechanisms**



# Transport vs Tunnel Mode

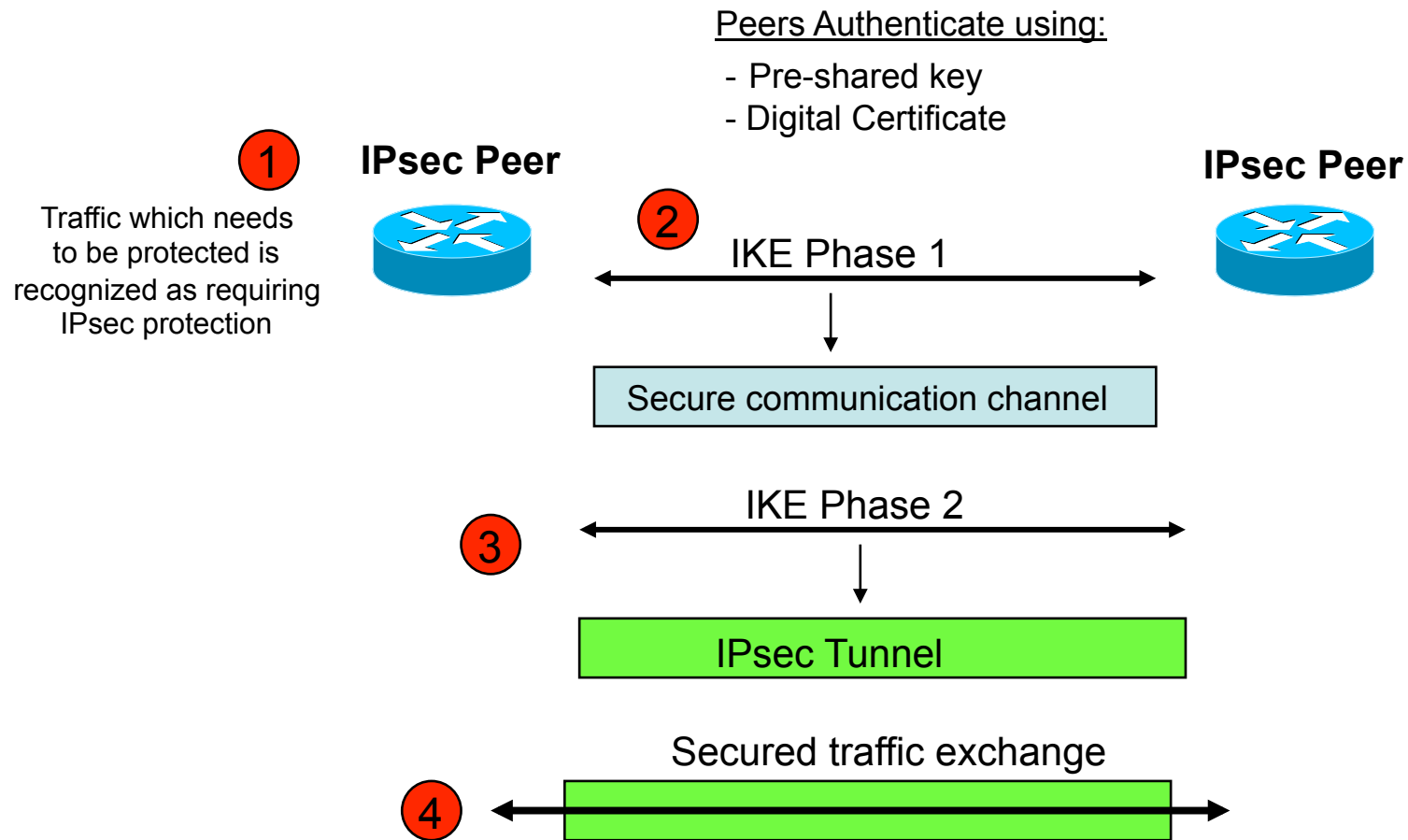


**Transport Mode:** End systems are the initiator and recipient of protected traffic

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

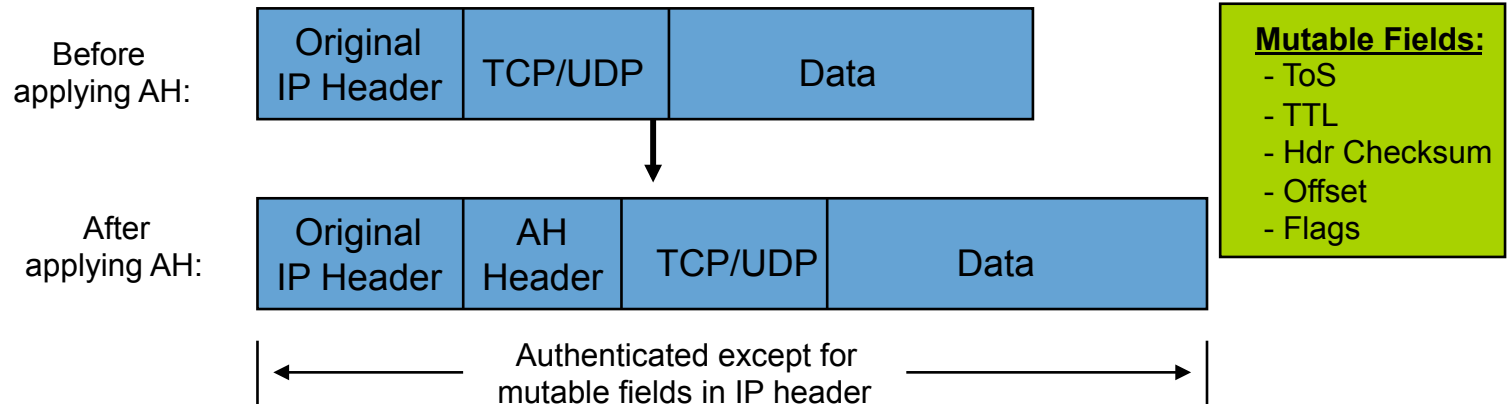


# IPsec with IKE

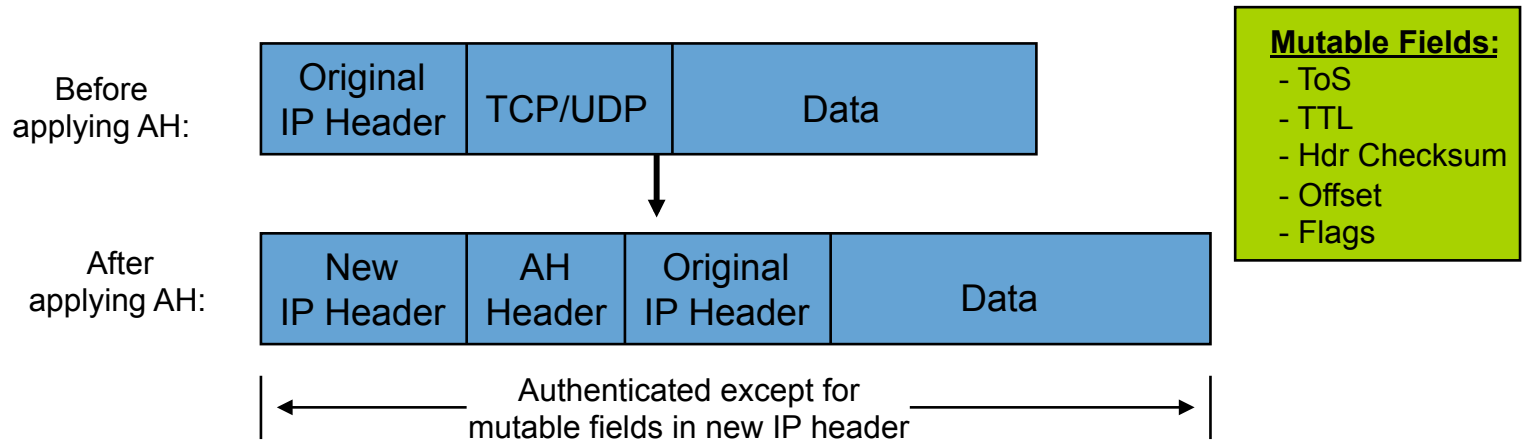


# IPv4 IPsec AH

## IPv4 AH Transport Mode:

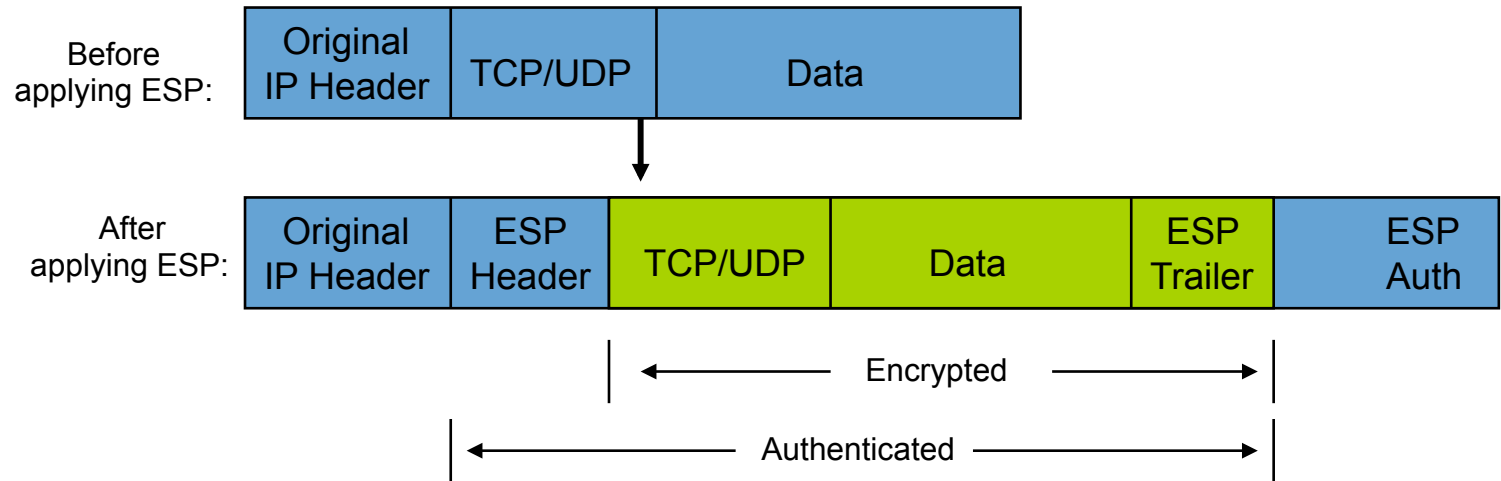


## IPv4 AH Tunnel Mode:

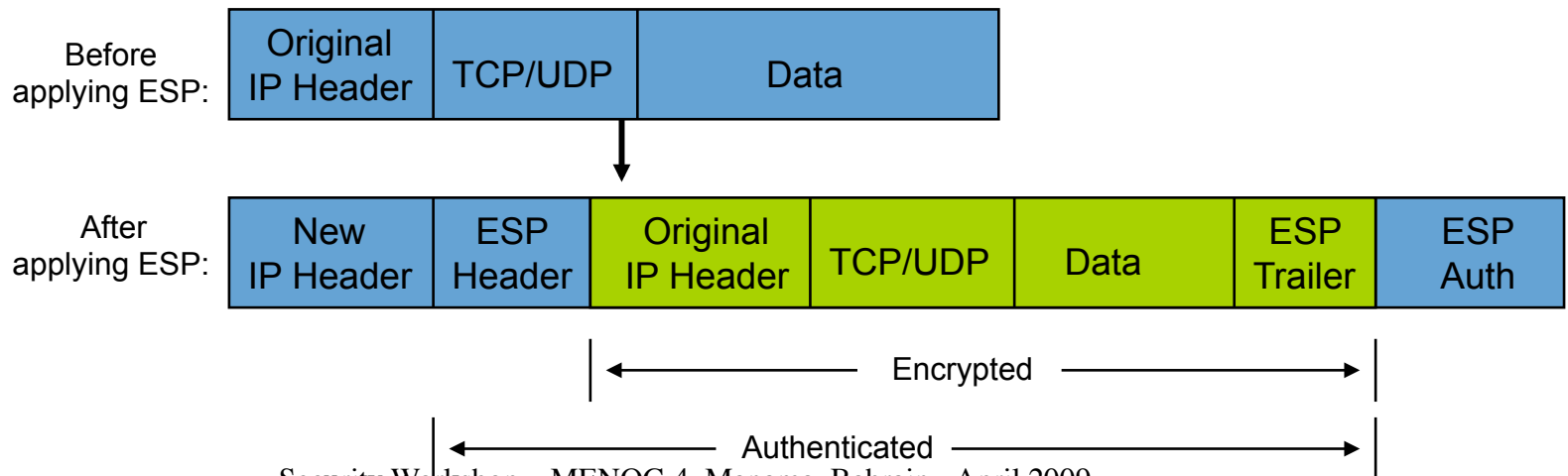


# IPv4 IPsec ESP

## IPv4 ESP Transport Mode:



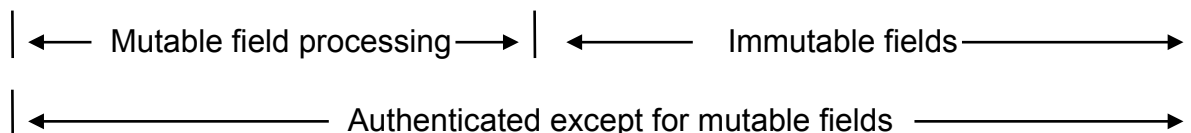
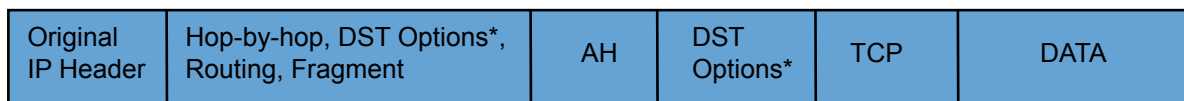
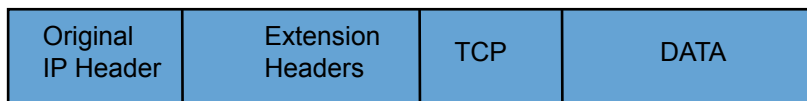
## IPv4 ESP Tunnel Mode:





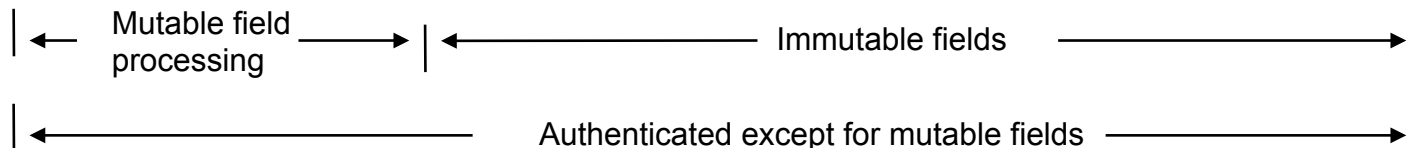
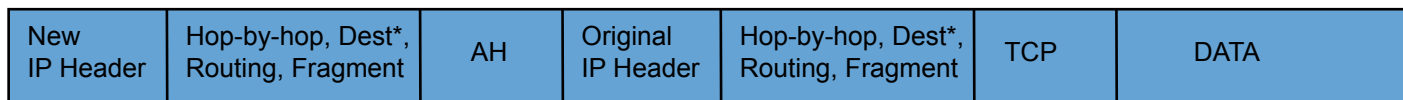
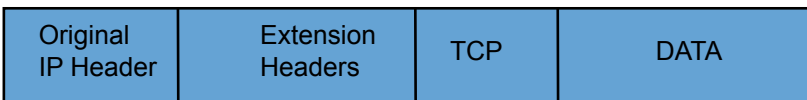
# IPv6 IPsec AH

## IPv6 AH Transport Mode:



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

## IPv6 AH Tunnel Mode:

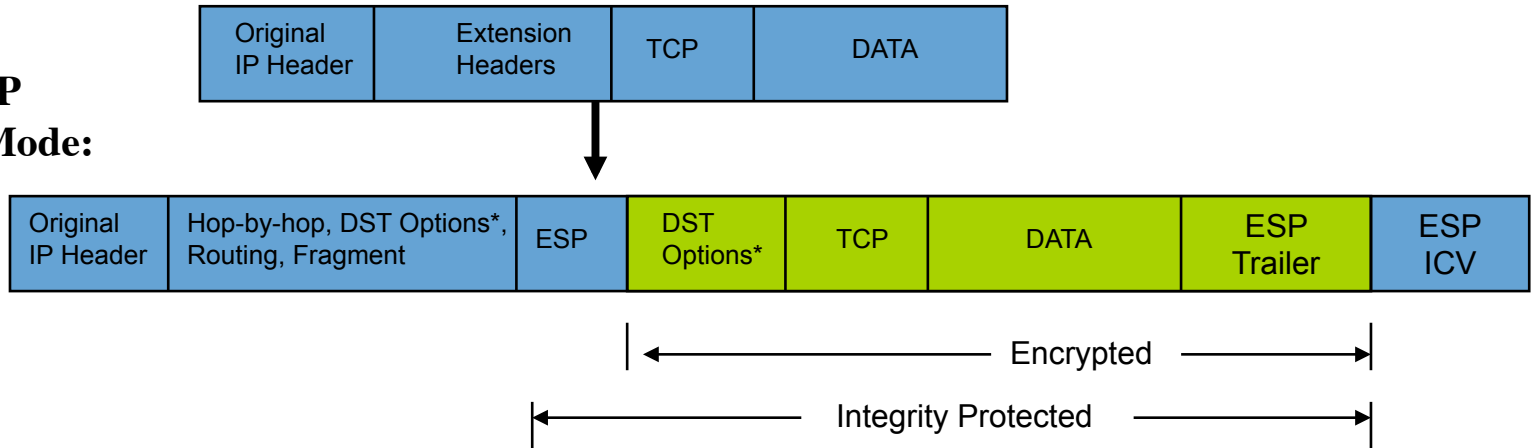


- Mutable Fields:**
- DSCP
  - ECN
  - Flow Label
  - Hop Limit

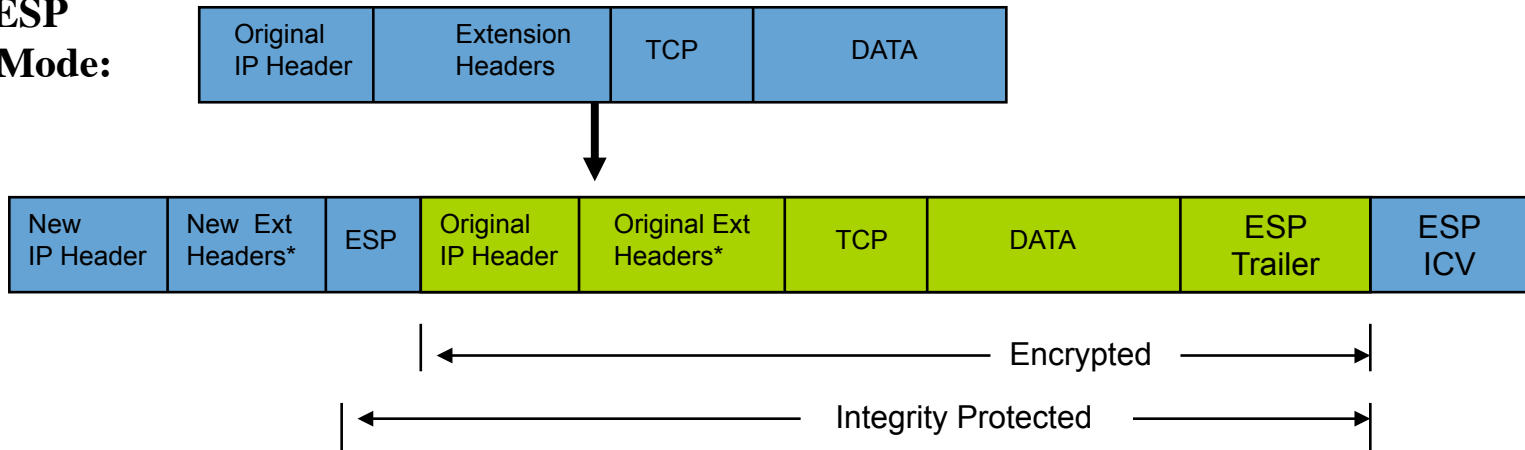


# IPv6 IPsec ESP

## IPv6 ESP Transport Mode:



## IPv6 ESP Tunnel Mode:

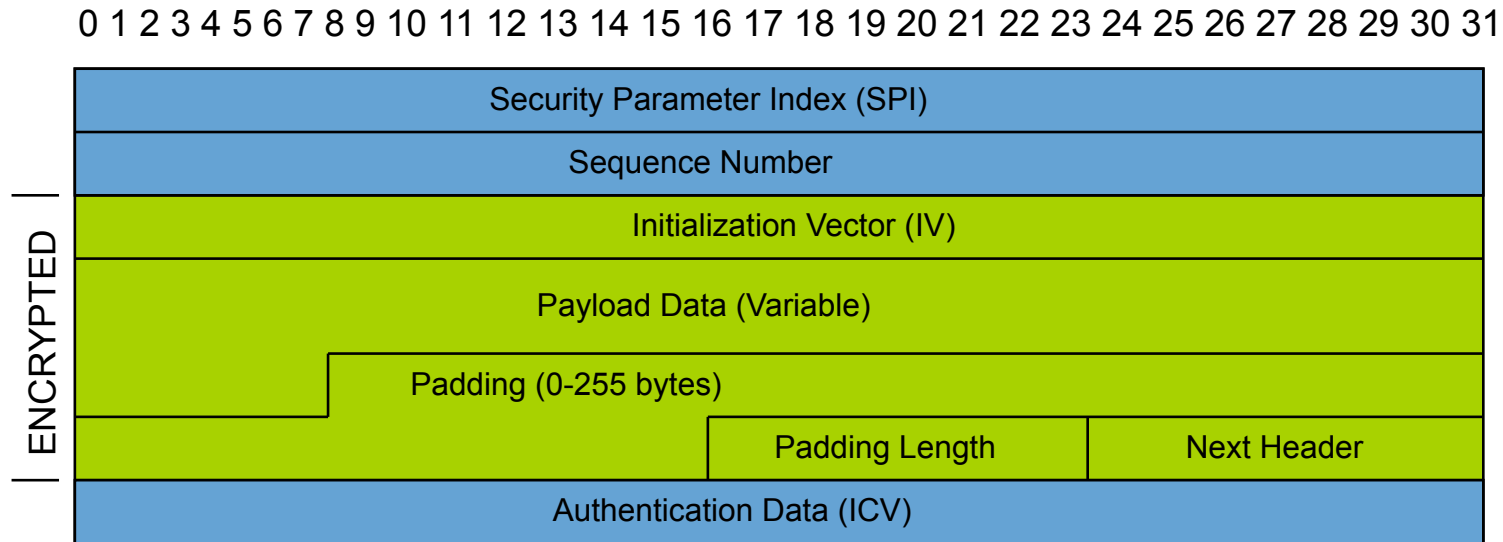


# IPv6 Enhancements Needed

- Standards Modifications
  - Need to take into consideration Stateless Autoconfiguration where Router Advertisement sends network prefix
  - Need to be able to differentiate between encrypted versus integrity protected traffic
- Usability
  - Interoperable defaults
  - Consistent terminology



# 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)



# 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



# 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

IKE Phase 1 SA

IKE SA

ISAKMP SA

Main Mode

## DH Key Length

DH Group

Modp #

Group #

## IKE Phase 2

IKE Phase 2 SA

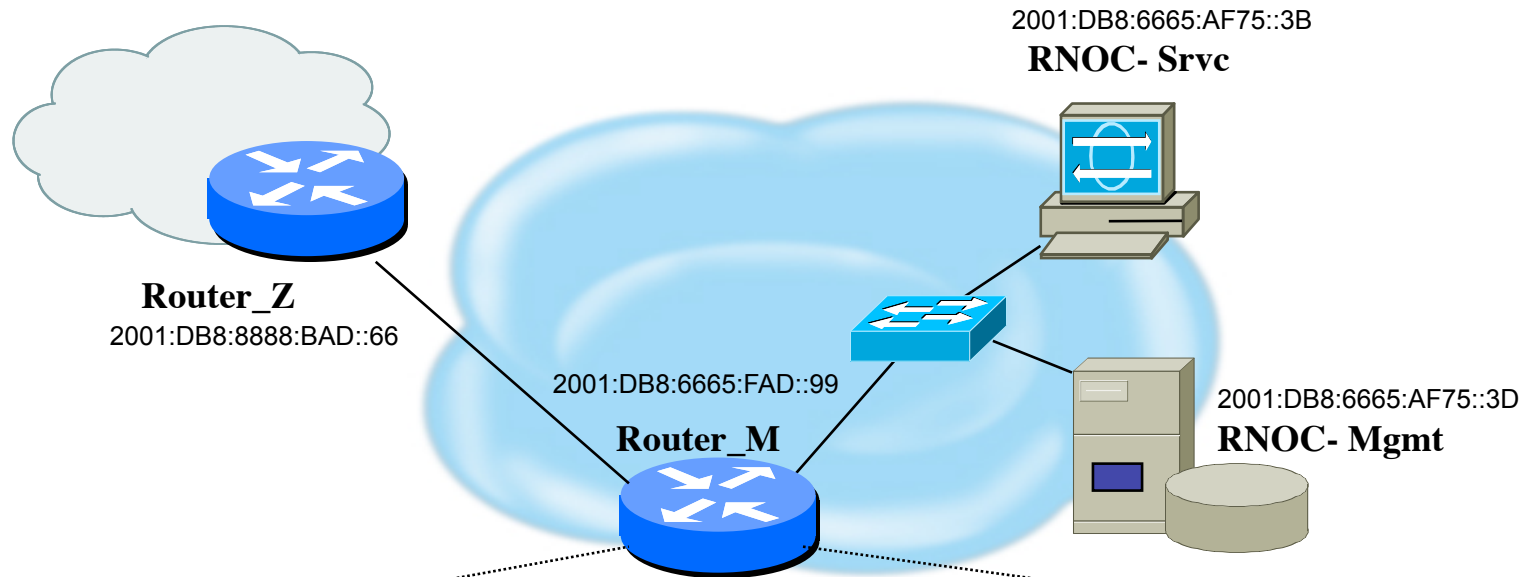
IPsec SA

Quick 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
  - AH authentication algorithm and keys
  - 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 ?



# Unix IPsec IKE Daemons

- Racoon2 (IKEv1 and IKEv2 and KINK)
  - <http://www.racoon2.wide.ad.jp/w/>
- Ipsec-tools (IKEv1)
  - port of KAME's IPsec utilities to the Linux-2.6 IPsec implementation; it supports NetBSD and FreeBSD as well
  - <http://ipsec-tools.sourceforge.net/>
- Strongswan (IKEv1 and IKEv2)
  - <http://www.strongswan.org/>
- Openikev2 (IKEv2)
  - <http://openikev2.sourceforge.net/>



# LINUX and MACOSX machines

- Type command ‘ *man racoon* ’
  - Read how to set-up racoon, the name for this particular IKE software
- Type command ‘ *man setkey* ’
  - This command is used to set up the SA database
- The following files are located in */etc/raccoon*:
  - ***psk.txt*** – file which contains the shared secrets
  - ***raccoon.conf*** – file which configures IKE phase 1 and IKE phase 2 parameters



# Set Up Security Policy Database

- Create a file named '*ipsec.conf*' which will be used with *setkey* to establish the correct security associations. The file should have the following information:
  - *flush;*
  - *spdflush;*
  - *spdadd 2001:DB8:6665:AF75::3D/128  
2001:DB8:8888:BAD::66/128 any -P out ipsec esp/  
transport//require ;*
  - *spdadd 2001:DB8:8888:BAD::66/128  
2001:DB8:6665:AF75::3D/128 any -P in ipsec esp/  
transport//require ;*



# Creating SA Database

- Test to see what happens when you try and create an SA database:
- Type the following:
  - `setkey -f /etc/racoon/ipsec.conf`
- Use the ‘ `setkey -P -D` ’ command to see if appropriate entries have been created



# Pre-Shared Key Configuration

- Edit the psk.txt file to add the peer IP address and the pre-shared secret key:

```
- # file for pre-shared keys used for IKE authentication
- # format is: 'identifier' 'key'
- # For example:
- # 10.1.1.1          flibbertigibbet
- # www.example.com  12345
- # foo@www.example.com micropachycephalosaurus
- <peer IPv6 address> <shared secret>
```

- Since the psk.txt file contains sensitive information make sure that the file is appropriately protected:

```
- chmod 600 /etc/racoon/psk.txt
```



# Operational Security Summary

- No single technique will solve all problems
- Ensure the simple sanity checks are in place
- Enforce filtering policies
- Log filter exceptions and monitor them
- Make friends with your upstream and downstream neighbors and support each other during DoS attacks

