Hardening IPv6 Network Devices

ISP Workshops

Agenda

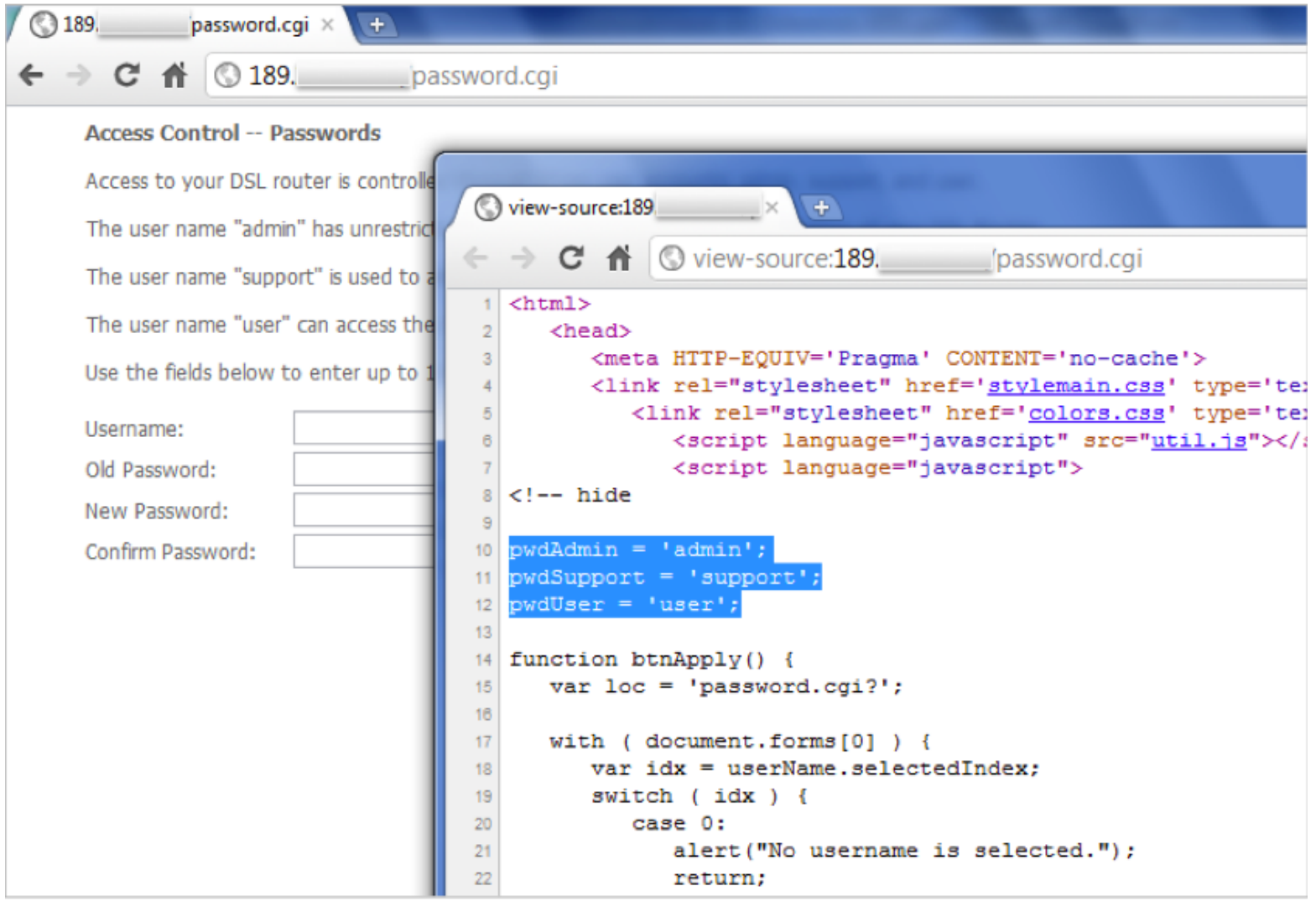
- Limiting Device Access
- Secure SNMP Access
- Securing the Data Path
- Configuration and Archiving
- Threats against Routing Protocols

Limiting Device Access

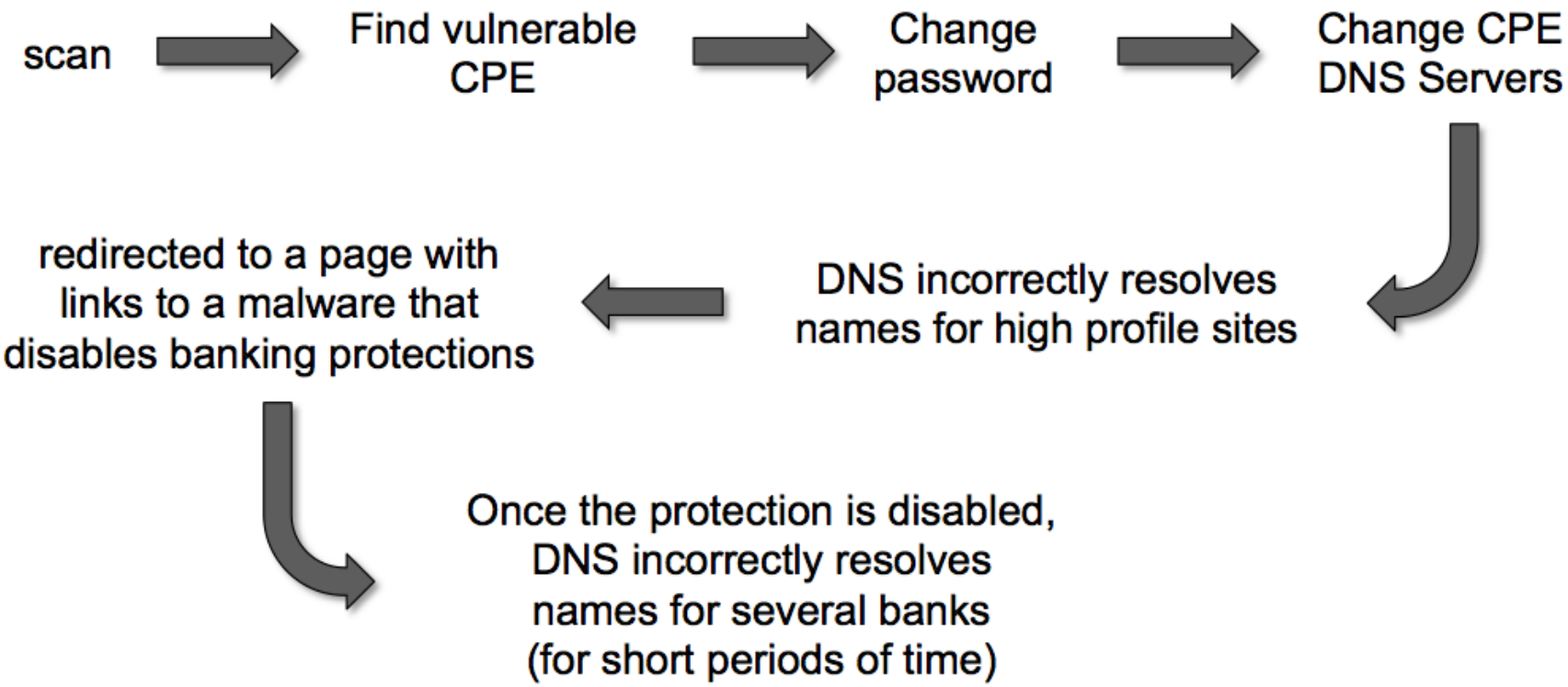
Think of ALL Devices

- The following problem was reported in 2013 and affects low-end CPEs (ADSL connections only)
  - Admin password exposed via web interface
  - Allow WAN management (this means anyone on Internet)
  - Bug fixed and reintroduced depending on the firmware version
- The bug is quite a number of years old

Password Visible via Web Interface



How CPE are Exploited



Magnitude of Problem

- 4.5 Million CPEs (ADSL Modems) using a unique malicious DNS
- In early 2012 more than 300,000 CPEs still infected
- 40 malicious DNS servers found

- Could device hardening have made a difference?

Device Physical Access

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

- "If you can touch it… the device now belongs to you"

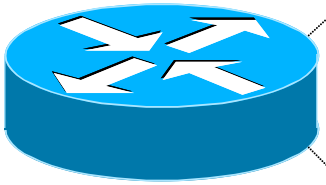
Interface Hardening

- IPv4
  - no ip proxy-arp
  - no ip unreachables
  - no ip redirects
  - no ip directed-broadcast
  - no ip mask-reply
- IPv6
  - no ipv6 unreachables
  - no ipv6 redirects

Device Access Control

- 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
- IF AVAILABLE – use digital certificate based authentication mechanisms instead of passwords

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 0
!
enable secret enable-secret
username dean secret dean-secret
```

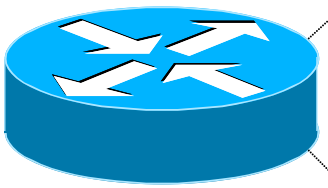
Never Leave Passwords in Clear-Text

- 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 dean secret dean-secret
username miwa secret miwa-secret
username pfs secret pfs-secret
username staff secret group-secret
```

Do NOT have group passwords!

User Authentication: Good

- From Cisco IOS 12.3, MD5 encryption was added for user passwords
  - **Do NOT use type 7 encryption**
    - (it is easy to reverse)

```
aaa new-model
aaa authentication login neteng local
username pfs secret 5 $1$j6Ac$3KarJszBV3VMaL/2Nio3E.
username dean secret 5 $1$LPV2$Q04NwAudy0/4AHHHQHvWj0
line vty 0 4
 login neteng
 access-class 3 in
```

User Authentication: Better

- Use centralised authentication system
  - RADIUS (not recommended for system security)
  - TACACS+

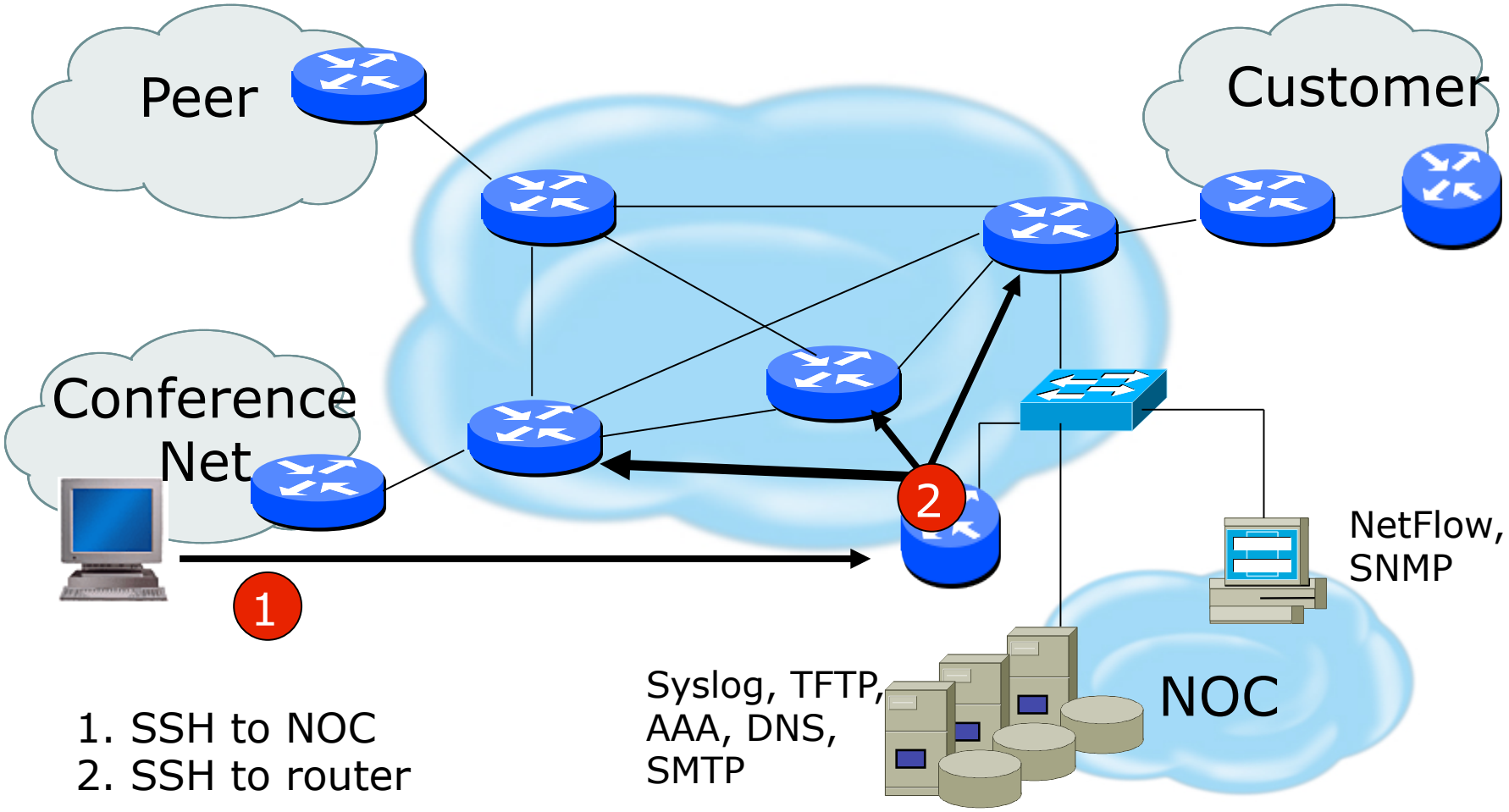
```
aaa new-model
aaa authentication login default group tacacs+ enable
aaa authentication enable default group tacacs+ enable
aaa accounting exec start-stop group tacacs+
!
ip tacacs source-interface Loopback0
tacacs server IPv6-TP
 address ipv6 2001:DB8::1
 key CKr3t#
tacacs server IPv4-TP
 address ipv4 192.168.1.1
 key CKr3t#
line vty 0 4
 access-class 3 in
```

Restrict Access To Trusted Hosts

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

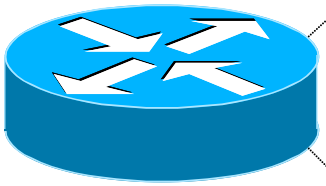
```
ip access-list extended VTY
 permit tcp host 192.168.200.7 192.168.1.0 0.0.0.255 eq 22 log-input
 permit tcp host 192.168.200.8 192.168.1.0 0.0.0.255 eq 22 log-input
 permit tcp host 192.168.100.6 192.168.1.0 0.0.0.255 eq 23 log-input
 deny   ip any any log-input
!
line vty 0 4
 access-class VTY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

Using an SSH 'Jumphost'



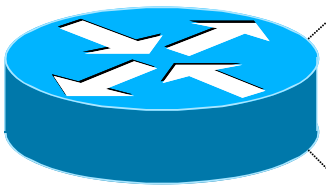
1. SSH to NOC
2. SSH to router

Banner – What Is Wrong ?



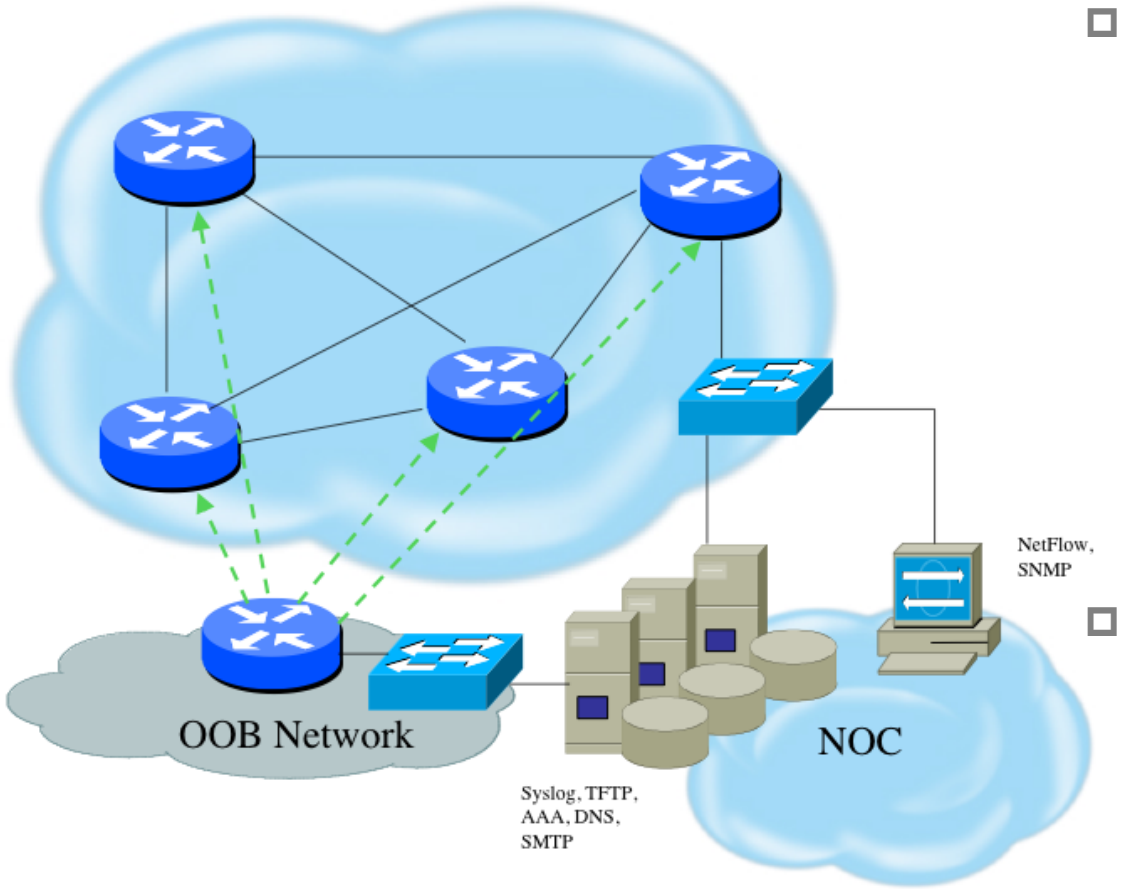
```
banner login ^C

      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.
```

Device OOB Management



- Out-of-band device management should be used to ensure DoS attacks do not hinder getting access to critical infrastructure devices
- Dial-back encrypted modems are sometimes still used as backup

Device Management Common Practice (1)

- SSH used exclusively
  - Do NOT use Telnet, not even from Jumphosts
- HTTP and HTTPS 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

Device Management Common Practice (2)

- 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

Turn Off Unused Services

- Global Services
  - no service finger (before Cisco IOS 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

Secure SNMP Access

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
```

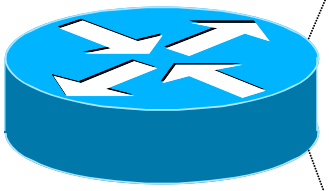
- If the router has SNMP, protect it!

```
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
```

Secure SNMP Access



```
ipv6 access-list SNMP-PERMIT
 permit ipv6 2001:DB8:22::/64 any
 permit ipv6 any 2001:DB8:22::/64
!
no snmp community public
no snmp community private
!
snmp-server enable traps
snmp-server enable traps snmp authentication
snmp-server enable traps snmp coldstart
snmp-server trap-source Loopback0
snmp-server community v6comm RO ipv6 SNMP-PERMIT
```

SNMP Best Practices

- Do not enable read/write access unless really necessary
  - Read – for access by Networking Monitoring System (eg LibreNMS)
  - Write – never!
- Choose community strings that are difficult to guess
  - Use same algorithm as for passwords
- Limit SNMP access to specific IP addresses
- Limit SNMP output with views

Secure Logging Infrastructure

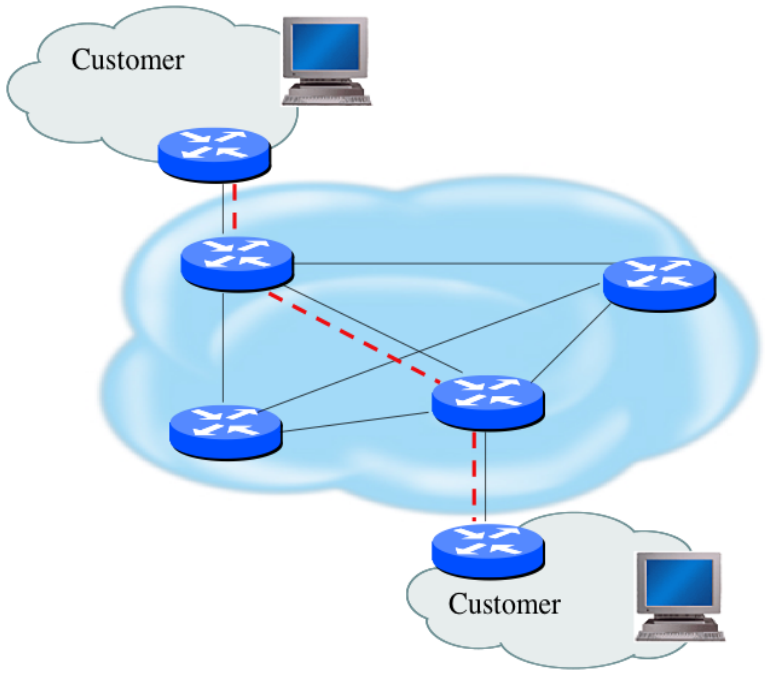
- Log enough information to be useful but not overwhelming.
- Create backup plan for keeping track of logging information should the syslog server be unavailable
- Remove private information from logs
- How accurate are your timestamps?
  - NTP needs to be configured
  - Synchronise with trusted time sources, eg pool.ntp.org or GPS receivers

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 Path

Securing The Data Path



- Filtering and rate limiting are primary mitigation techniques
- Edge filter guidelines for ingress filtering (BCP38/ BCP84)
- Null-route and black-hole any detected malicious traffic
- Netflow is primary method used for tracking traffic flows
- 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?

General Filtering Best Practices

- 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 Filtering

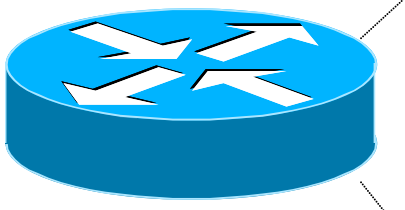
```
ipv6 access-list INBOUND-iACL
 remark Permit the legitimate signaling traffic (BGP, EIGRP, PIM)
 permit tcp host 2001:db8:20::1 host 2001:db8:20::2 eq bgp
 permit tcp host 2001:db8:20::1 eq bgp host 2001:db8:20::2
 permit 88 any any
 permit 103 any any
 remark Permit NDP packets
 permit icmp any any nd-na
 permit icmp any any nd-ns
 permit icmp any any router-advertisement
 permit icmp any any router-solicitation
 remark Deny RH0 and other unknown extension headers
 deny ipv6 any any routing-type 0 log
 deny ipv6 any any log undetermined-transport
 remark Permit the legitimate management traffic
 permit tcp 2001:db8:11::/48 any eq 22
 permit tcp 2001:db8:11::/48 any eq www
 permit udp 2001:db8:11::/48 any eq snmp
 remark Deny any packets to the infrastructure address space
 deny ipv6 any 2001:db8:2222::/48
 deny ipv6 any 2001:db8:20::/48
 permit ipv6 any any
!
interface FastEthernet 0/0
 description Connection to outside network
 ipv6 address 2001:db8:20::2/64
 ipv6 traffic-filter INBOUND-iACL in
```

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.

But What About Egress Filtering?

- 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)



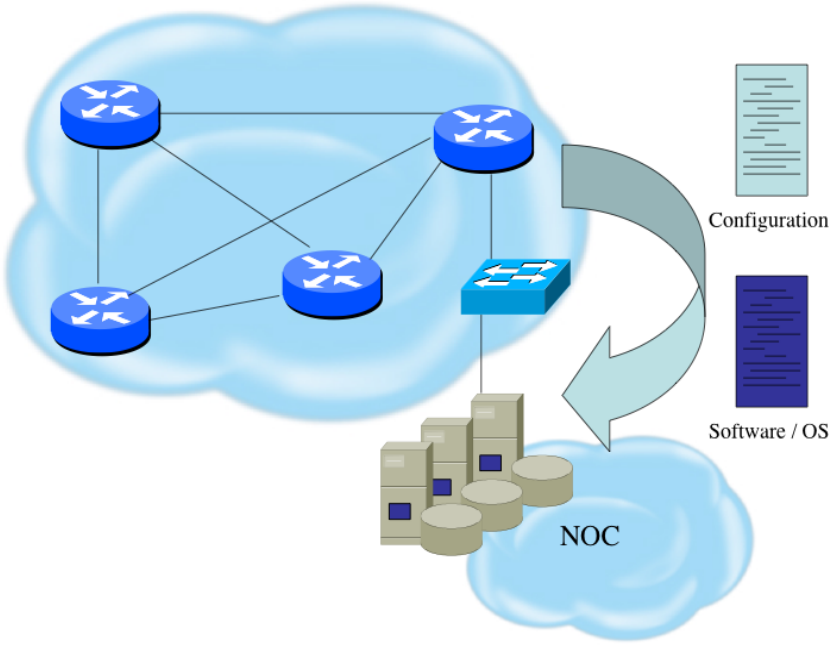
```
ipv6 access-list DSL-ipv6-Outbound
 permit ipv6 2001:DB8:AA65::/48  any
 deny   ipv6 any any log

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

Configuration and archiving

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 and Configuration Upgrade / Integrity



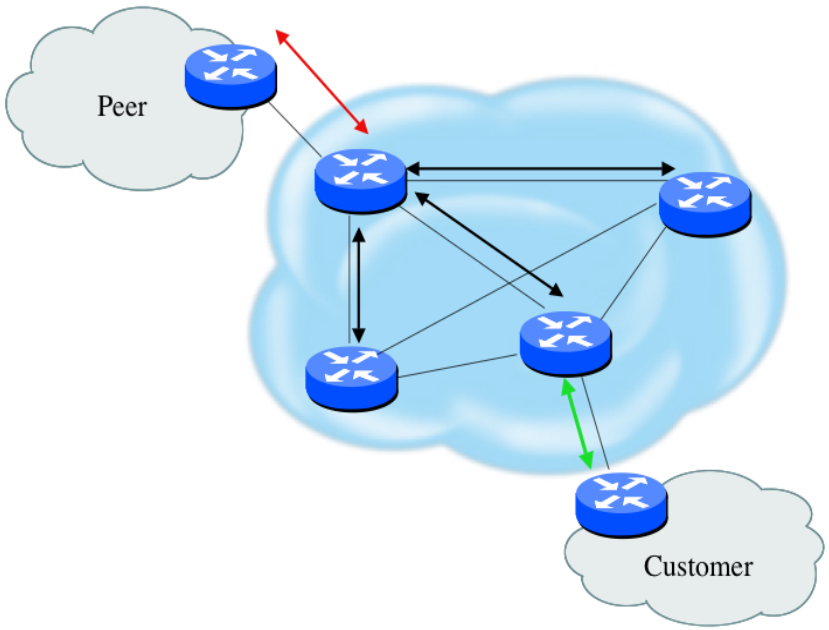
- Files stored on specific systems with limited access
- All access to these systems are authenticated and audited
- SCP is used where possible; FTP is NEVER used; TFTP still used
- Configuration files are polled and compared on an hourly basis (RANCID)
- Filters limit uploading / downloading of files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

Threats Against Routing Protocols

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 for 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

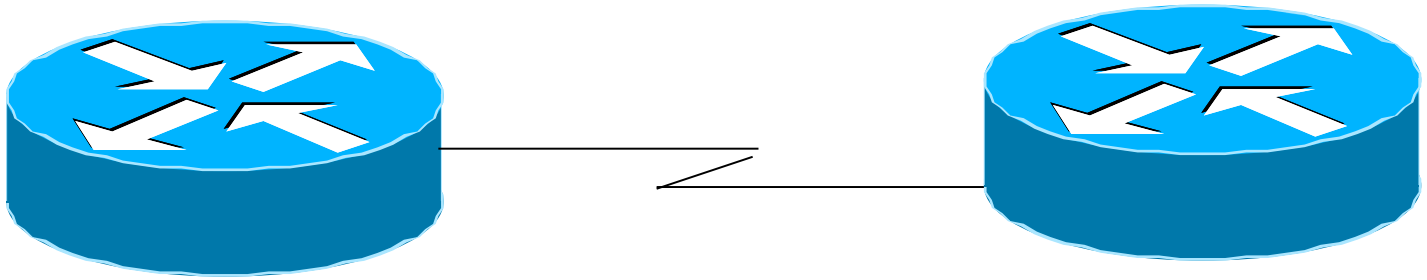


- MD-5 authentication
  - Some deploy at customer's request
- Route filters limit what routes are believed from a valid peer
- Packet filters limit which systems can appear as a valid peer
- Limiting propagation of invalid routing information
  - Prefix filters
  - AS-PATH filters (trend is leaning towards this)
  - Route damping (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….

Sample MD-5 Auth Configuration (OSPFv2)



```
interface Loopback0
 ip address 70.70.70.70 255.255.255.255
 ip ospf 10 area 0
!
interface Serial2
 ip address 192.16.64.2 255.255.255.0
 ip ospf 10 area 0
 ip ospf message-digest-key 1 md5 mk6
!
router ospf 10
 area 0 authentication message-digest
```

```
interface Loopback0
 ip address 172.16.10.36 255.255.255.255
 ip ospf 10 area 0
!
interface Serial1/0
 ip address 192.16.64.1 255.255.255.0
 ip ospf 10 area 0
 ip ospf message-digest-key 1 md5 mk6
!
router ospf 10
 area 0 authentication message-digest
```

Sample OSPFv3 IPSec Configuration

```
interface Loopback0
 ipv6 address 2001:DB8::1/128
 ipv6 ospf 100 area 0

interface FastEthernet0/0
 description Area 0 backbone interface
 ipv6 address 2001:DB8:2000::1/64
 ipv6 ospf 100 area 0

interface FastEthernet0/1
 description Area 1 interface
 ipv6 address 2001:DB8:1000::2/64
 ipv6 ospf 100 area 1
 ipv6 ospf authentication ipsec spi 257 sha1 7 091C1E59495546435A5D557879767A6166714054455755
020D0C06015B564D400F0E01050502035C0C

ipv6 router ospf 100
 router-id 10.0.0.1
 log-adjacency-changes detail
 passive-interface Loopback0
 timers spf 0 1
 timers pacing flood 15
 area 0 range 2001:DB8::/64
 area 0 range 2001:DB8:2000::/64
 area 1 range 2001:DB8:1000::/64
 area 0 encryption ipsec spi 256 esp aes-cbc 256 7 075F711C1E59495547435A5D557B7A757961677041
55445153050A0B00075D504B420D0C03070601005E0E53520D02514650520D5D5059771A195E4E5240455C5B sha1
7 00544356540B5B565F701D1F5848544643595E567879767A6166714054455052050D0C07005A574C42
```

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 Security Techniques

- BGP Community Filtering
- MD5 Keys on the eBGP and iBGP Peers
- Max Prefix Limits
- Max AS Path Length
- Prefer Customer Routes over Peer Routes (RFC 1998)
- GTSM (i.e. TTL Hack)
- Remote Trigger Black Hole (RTBH) Filtering

BGP Community Filtering

- Network operators use BGP Communities for:
  - Internal policies
  - Policies for their customers
  - Policies towards their upstream providers
- Policies are aimed at ensuring routing system integrity within networks and between networks
- BGP Community references:
  - Specification (RFC1997) and Example Use (RFC1998)
  - http://bgp4all.com/ftp/isp-workshops/BGP\Presentations/09-BGP-Communities.pdf

MD5 keys on BGP peerings

- Use passwords on all BGP sessions
  - Not being paranoid, VERY necessary
  - It's a secret shared between you and your peer
  - If arriving packets don't have the correct MD5 hash, they are ignored
  - Helps defeat miscreants who wish to attack BGP sessions
- Powerful preventative tool, especially when combined with filters and GTSM

```
router bgp 100
 address-family ipv6
  neighbor 2001:db8::1 remote-as 200
  neighbor 2001:db8::1 description Peering with AS200
  neighbor 2001:db8::1 password 7 030752180500
!
```

BGP Maximum Prefix Tracking

- Allow configuration of the maximum number of prefixes a BGP router will receive from a peer
- Two level control:
  - Warning threshold: log warning message
  - Maximum: tear down the BGP peering, manual intervention required to restart

```
neighbor <x.x.x.x> maximum-prefix <max> [restart N] [<threshold>] [warning-only]
```

- Optional keywords:
  - **`restart`** will restart the BGP session after N minutes
  - <threshold> sets the warning level (default 75%)
  - **`warning-only`** only sends warnings

Limiting AS Path Length

- Some BGP implementations have problems with long AS_PATHS
  - Memory corruption
  - Memory fragmentation
- Even using AS_PATH prepends, it is not normal to see more than 20 ASes in a typical AS_PATH in the Internet today
  - The Internet is around 5 ASes deep on average
  - Largest AS_PATH is usually 16-20 ASNs

```
neighbor x.x.x.x maxas-limit 15
```

Limiting AS Path Length

- Some announcements have ridiculous lengths of AS-paths:

```
*> 3FFE:1600::/24      22 11537 145 12199 10318 10566 13193 1930 2200
   3425 293 5609 5430 13285 6939 14277 1849 33 15589 25336 6830 8002
   2042 7610 i
```

This example is an error in one IPv6 implementation

```
*>i193.105.15.0        2516 3257 50404 50404 50404 50404 50404 50404
   50404 50404 50404 50404 50404 50404 50404 50404 50404 50404 50404
   50404 50404 50404 50404 50404 50404 50404 50404 50404 50404 50404
   50404 50404 50404 50404 50404 50404 50404 50404 50404 50404 50404
   50404 50404 50404 50404 50404 50404 50404 50404 50404 50404 50404
   50404 50404 50404 50404 50404 50404 50404 50404 50404 50404 50404
   50404 50404 50404 50404 50404 50404 50404 50404 50404 50404 50404
   50404 50404 50404 50404 50404 50404 50404 50404 50404 50404 50404
   50404 50404 50404 50404 50404 50404 50404 i
```

This example shows 100 prepends (for no obvious reason)

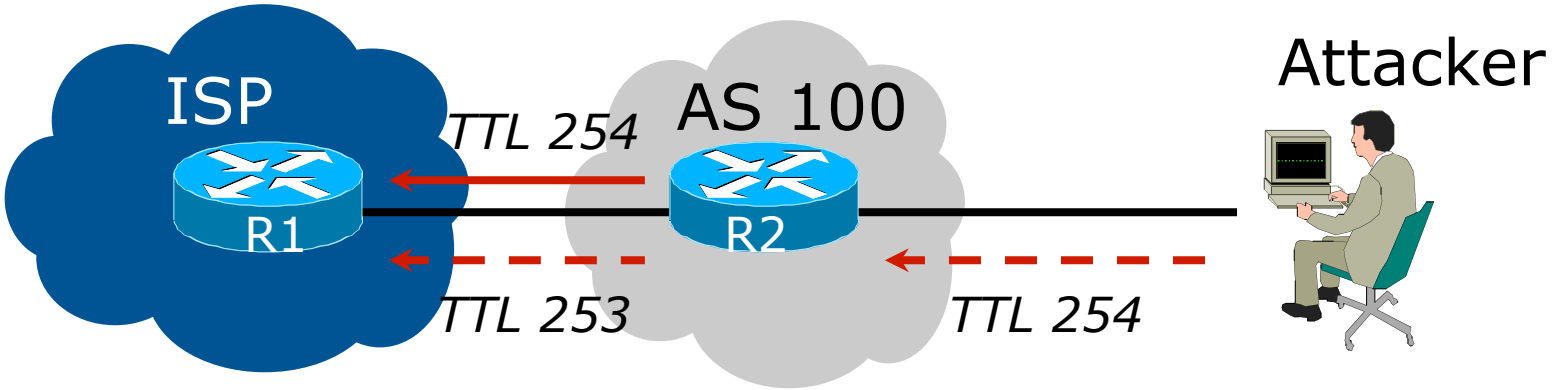
- If your implementation supports it, limit the maximum AS-path length you will accept

Customer routes vs Peer routes

- Common for end organisations to have more than one upstream provider
- Routes heard from the customer have to be preferred over the same routes heard from a peer
  - This is done by increasing BGP local preference for customer routes
  - Provides a degree of protection for its customer routes

GTSM: The BGP TTL "hack"

- Implement RFC5082 on BGP peerings
  - (Generalised TTL Security Mechanism)
  - Neighbour sets TTL to 255
  - Local router expects TTL of incoming BGP packets to be 254
  - No one apart from directly attached devices can send BGP packets which arrive with TTL of 254, so any possible attack by a remote miscreant is dropped due to TTL mismatch



BGP TTL "hack"

- TTL Hack:
  - Both neighbours must agree to use the feature
  - TTL check is much easier to perform than MD5
  - (Called BTSH – BGP TTL Security Hack)
- Provides "security" for BGP sessions
  - In addition to packet filters of course
  - MD5 should still be used for messages which slip through the TTL hack
  - See https://www.nanog.org/meetings/nanog27/presentations/meyer.pdf for more details

Remotely Triggered Black Hole Filtering

- A simple technique whereby the Network Operator can use their entire backbone to block mischievous traffic to a specific address within their network or their customers' network
- Chris Morrow's presentation at NANOG 30 in 2004 describing the technique:
  - https://www.nanog.org/meetings/nanog30/presentations/morrow.pdf
- Deployed and supported by many of the world's largest network operators

RTBH – How it works

- Network Operator deploys:
  - RTBH support across their entire backbone
    - Simply a null route for a specific next-hop address
    - (Router Null interfaces simply discard packets sent to them – negligible overhead in modern hardware)
  - A trigger router (usually in the NOC)
    - Talks iBGP with the rest of the backbone (typically as a client to route-reflectors in the core)
    - Used to trigger a blackhole route activity for any address under attack, as requested by a customer

RTBH – Backbone Configuration

- Network Operator sets up a null route for the 100::1 address on all the backbone routers which participate in BGP

```
ipv6 route 100::1/128 null 0  254
```

- 100::1 is part of 100::/64, the Discard Prefix, one of the reserved IPv6 address blocks listed in the IANA registry
  - http://www.iana.org/assignments/iana-ipv6-special-registry
  - It is not used or routed on the public Internet

RTBH – Trigger Router (1)

- Create a route-map to catch routes which need to be blackholed
  - Static routes can be tagged in Cisco IOS – we will tag routes to be blackholed with the value of 66
  - Set origin to be iBGP
  - Set local-preference to be 150
    - higher than any other local-preference set in the backbone
  - Set community to be *no-export* and internal marker community (ASN:666)
    - Don’t want prefix to leak outside the AS
  - Set next-hop to 192.0.2.1 (IPv4) or 100::1 (IPv6)

RTBH – Trigger Router (2)

- The whole route-map:

```
route-map v6blackhole-trigger permit 10
 description Look for Route 66
 match tag 66
 set local-preference 200
 set origin igp
 set community no-export 100:666
 set ip next-hop 100::1
!
route-map v6blackhole-trigger deny 20
 description Nothing else gets through
```

RTBH – Trigger Router (3)

- Then introduce the route-map into the BGP configuration
  - NB: the iBGP on the trigger router cannot use "next-hop-self" – Cisco IOS over writes the route-map originated next-hop with "next-hop-self"

```
router bgp 100
 address-family ipv6
  redistribute static route-map v6blackhole-trigger
  neighbor 2001:dbd::2 remote-as 100
  neighbor 2001:dbd::2 description iBGP with RR1
  neighbor 2001:dbd::2 update-source Loopback 0
  neighbor 2001:dbd::2 send-community
  neighbor 2001:dbd::3 remote-as 100
  neighbor 2001:dbd::3 description iBGP with RR2
  neighbor 2001:dbd::3 update-source Loopback 0
  neighbor 2001:dbd::3 send-community
!
```

RTBH – Trigger Router (4)

- To implement the trigger, simply null route whatever address or address block needs to be blackholed
  - With Tag 66

```
ipv6 route 2001:db8:f::e0/128 null0 tag 66
```

  - And this ensures that (for example) 2001:db8:f::e0/128 is announced to the entire backbone with next-hop 100::1 set

RTBH – End Result

- Prefixes which need to be null routed will come from the trigger router and look like this in the BGP table:

```
 *>i 2001:DB8:F::E0/128   100::1       0     200       0 i
```

- Routing entry for 2001:db8:f::e0 is this:

```
cr1>sh ipv6 route 2001:db8:f::e0
Routing entry for 2001:DB8:F::E0/128
  Known via "bgp 100", distance 200, metric 0, type
internal
  Route count is 1/1, share count 0
  Routing paths:
    100::1
      MPLS label: nolabel
      Last updated 00:00:03 ago
```

RTBH – End Result

- Routing entry for 100::1 is this:

```
cr1>sh ipv6 route 100::1
Routing entry for 100::1/128
  Known via "static", distance 1, metric 0
  Route count is 1/1, share count 0
  Routing paths:
    directly connected via Null0
      Last updated 00:05:21 ago
```

- Traffic to 2001:db8:f::e0 is sent to null interface

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

But Wait…There's More…

- RPKI – Resource Public Key Infrastructure, the Certificate Infrastructure to Support the other Pieces
  - We need to be able to authoritatively prove who owns an IP prefix and what AS(s) may announce it
  - Prefix ownership follows the allocation hierarchy (IANA, RIRs, ISPs, etc)
  - Origin Validation
    - Using the RPKI to detect and prevent mis-originations of someone else's prefixes (early 2012)
  - AS-Path Validation AKA BGPsec
    - Prevent Attacks on BGP (future work)

BGP – Why Origin Validation?

- Prevent YouTube accident & Far Worse
- Prevents most accidental announcements
- Does not prevent malicious path attacks
- That requires 'Path Validation' and locking the data plane to the control plane, the third step, BGPsec

Infrastructure Security Summary

- Every device in your network could be exploited so make sure to harden them all (especially change default username/passwords)
  - Printers, tablets, CPE's, etc
- Filtering help everyone – PLEASE deploy anti-spoofing filters
- Understand what you are sending in the clear from sending device to recipient and protect where needed
- Log and audit for trends since sometimes an abnormality can show the start of reconnaissance for a later attack

Hardening IPv6 Network Devices

ISP Workshops