



ISP and NSP Security Workshop

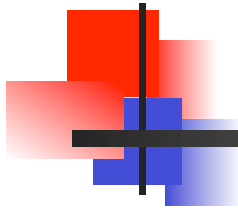
APRICOT 2009

18th February – 22nd February 2009



Free Use

- This slide deck can be used by any operator to help empower their teams, teach their staff, or work with their customers.
- It is part of the next generation of **APRICOT Security Curriculum** providing tools that can improve the quality of the Internet.



Goal

- Provide security core techniques/task that any SP can do to improve their resistance to security issues.
- These core techniques can be done on any core routing vendor's equipment.
- Each of these techniques have proven to make a difference.



Agenda

- Host Security
- Securing Infrastructure
- Gain Visibility
- MPLS / L3 VPN Security
- Understanding IPSec and SSL VPN
- Managed Security Services
- Conclusions



Time Table

- Day 1
 - Securing the Infrastructure
- Day 2 / 3
 - Gain Visibility
- Day 3
 - IPv6 Security
- Day 4 /5
 - MPLS / L3 VPN Security
- Day 5
 - Managed Security Services
- Conclusions



What Is Core Security?

- Often thought of as “SP Security”
 - What is an SP today?
- Internal networks are no longer truly internal
 - Tunneling
 - VPN
 - Worms, worms, worms
- The infrastructure is critical; if we can’t protect it, nothing else matters
 - Edge security initiatives abound: NAC, 802.1X, personal firewalls, etc.



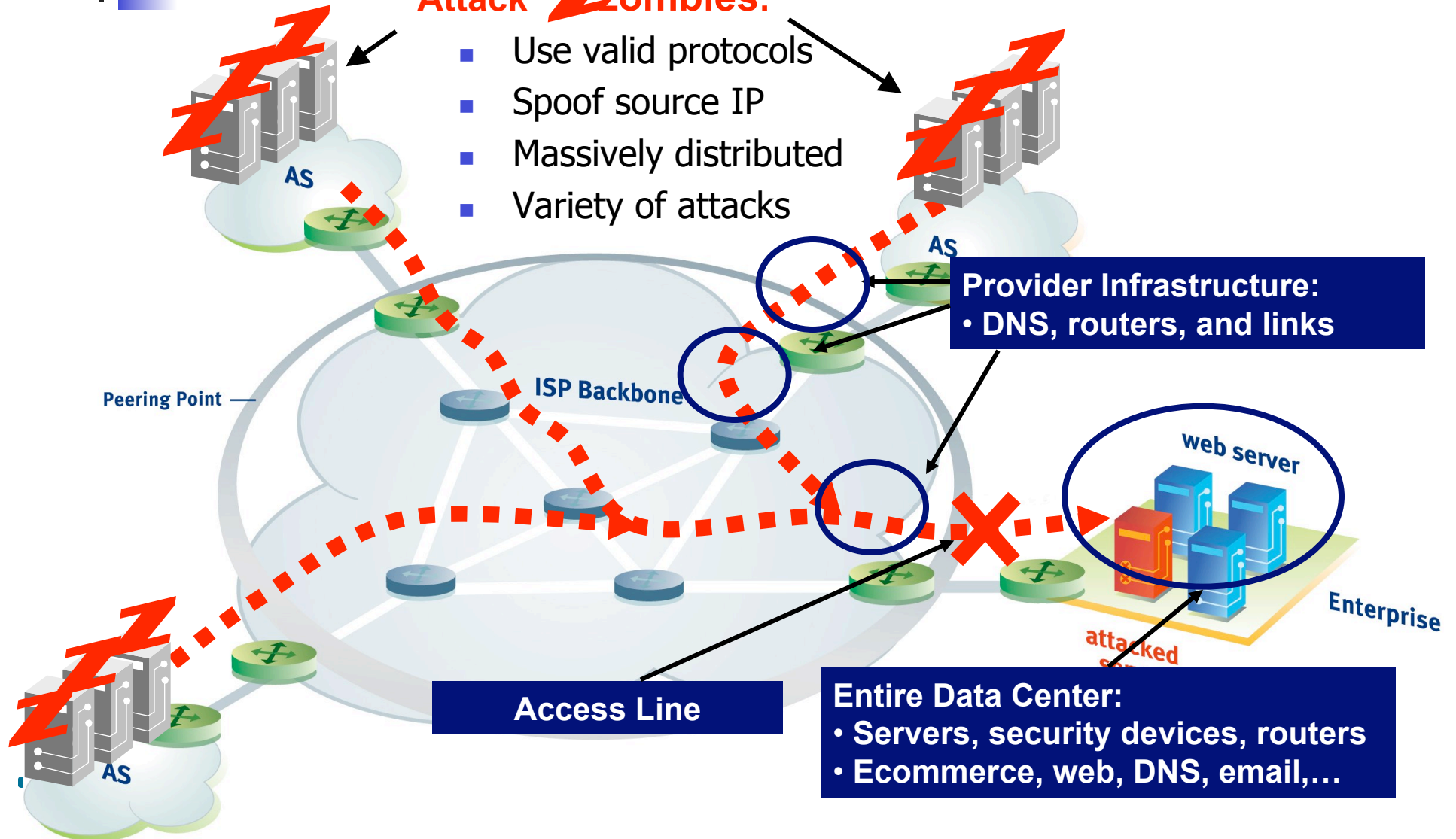
Denial of Service Attacks

- We understand intrusions (patch, patch, patch ;-))
- What about DoS? Do “the right things” and still suffer
- The vast majority of modern DoS attacks are distributed
 - DDos IS DoS
- DoS is often driven by financial motivation
 - DoS for hire :-)
 - Economically-driven miscreant community
- DoS cannot be ignored; your business depends on effective handling of attacks

DDoS Vulnerabilities, Threats and Targets

Attack **Zombies:**

- Use valid protocols
- Spoof source IP
- Massively distributed
- Variety of attacks

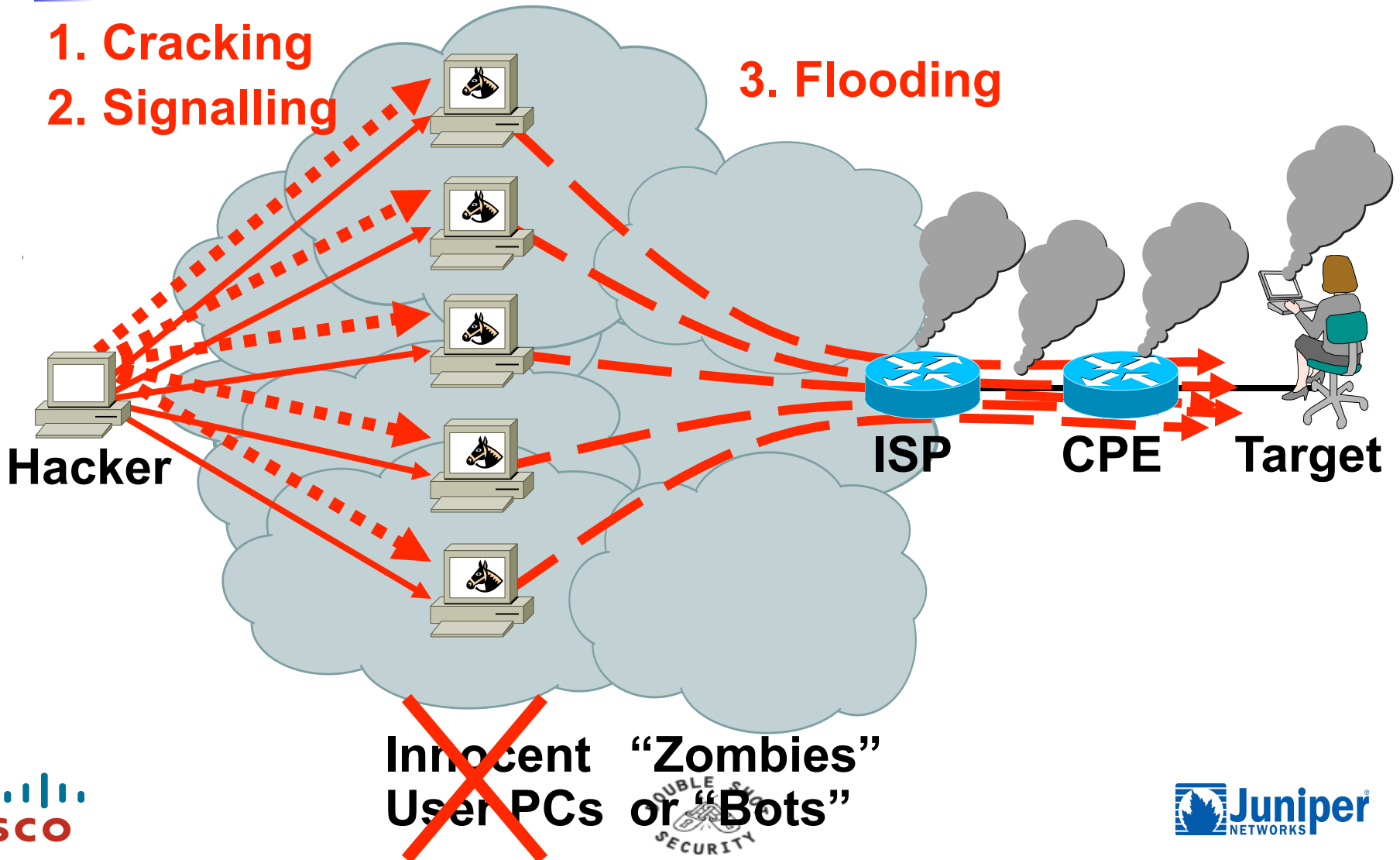


DoS: The Procedure

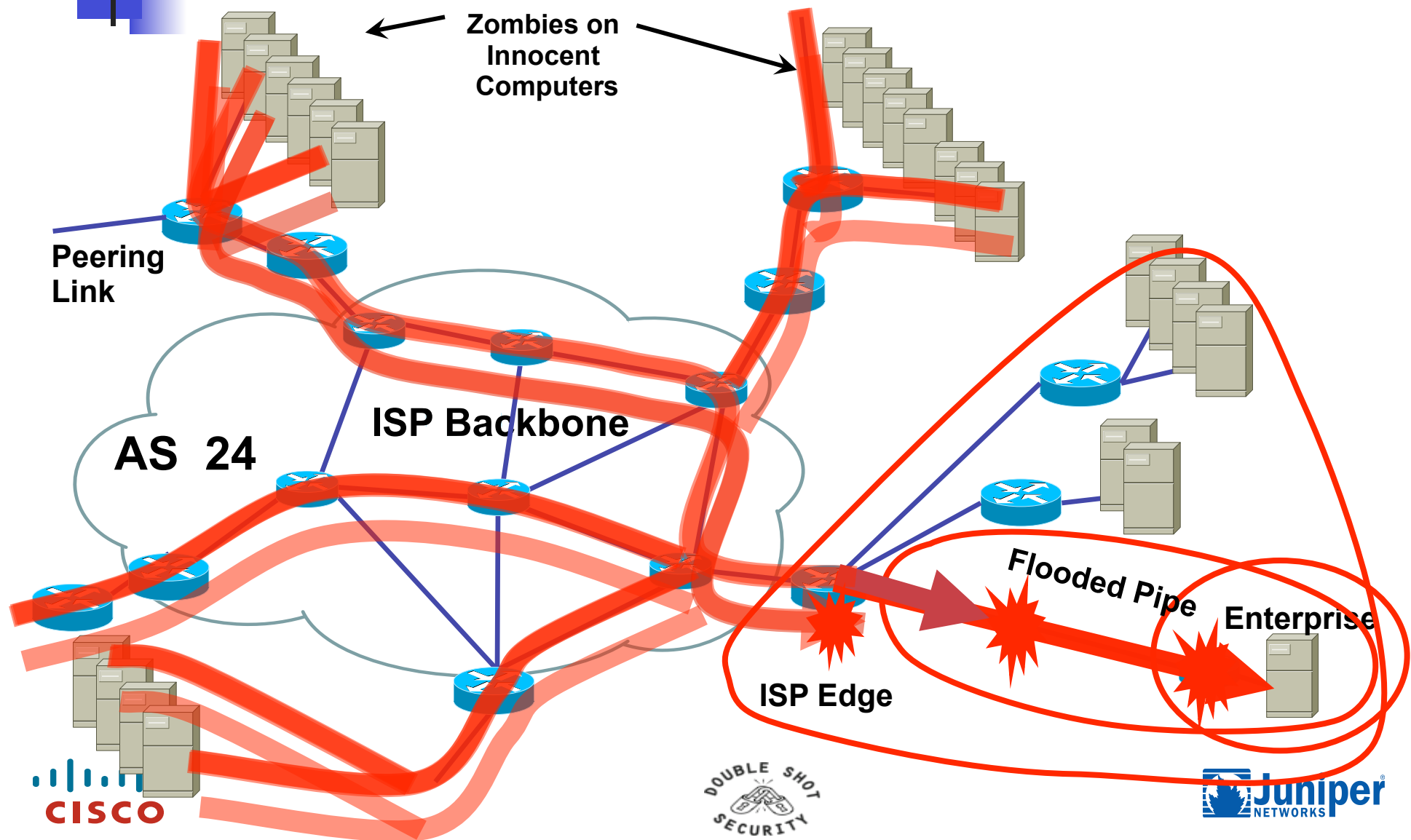
1. Cracking

2. Signalling

3. Flooding



An SP View: Denial of Service





Where to go to get more?

- **Operators Security Curriculum**

- Sessions recorded over time which builds a library for all SPs to use for their individual training, staff empowerment, and industry improvements.

- <http://www.nanog.org/ispsecurity.html>
- <http://www.apricot.net/apricot2005/workshop.html>



Infrastructure Attacks

- Infrastructure attacks are increasing in both volume and sophistication
 - Sites with Cisco documents and presentations on routing protocols (and I don't mean Cisco.com)
 - Marked increase in presentations about routers, routing and Cisco IOS vulnerabilities at conferences like Blackhat, Defcon and Hivercon
 - Router attack tools and training are being published
- Why mount high-traffic DDOS attacks when you can take out your target's gateway routers?
- Hijacked routers are valuable in the spam world, which has a profit driver
- Router compromise (0wn3d) due to weak password



From Bad to Worms

- Worms have emerged as the new security reality
- Old worms never die!
 - Millions of UPnP and Slammer packets still captured daily
- Most worms are intended to compromise hosts
- Worm propagation is dependant on network availability
- Worms and DoS are closely related
 - Secondary worm effects can lead to denial of service
 - Worms enable DoS by compromising hosts → BOTnets
- Perimeters are crumbling under the worm onslaught (VPN/mobile workers, partners, etc.)



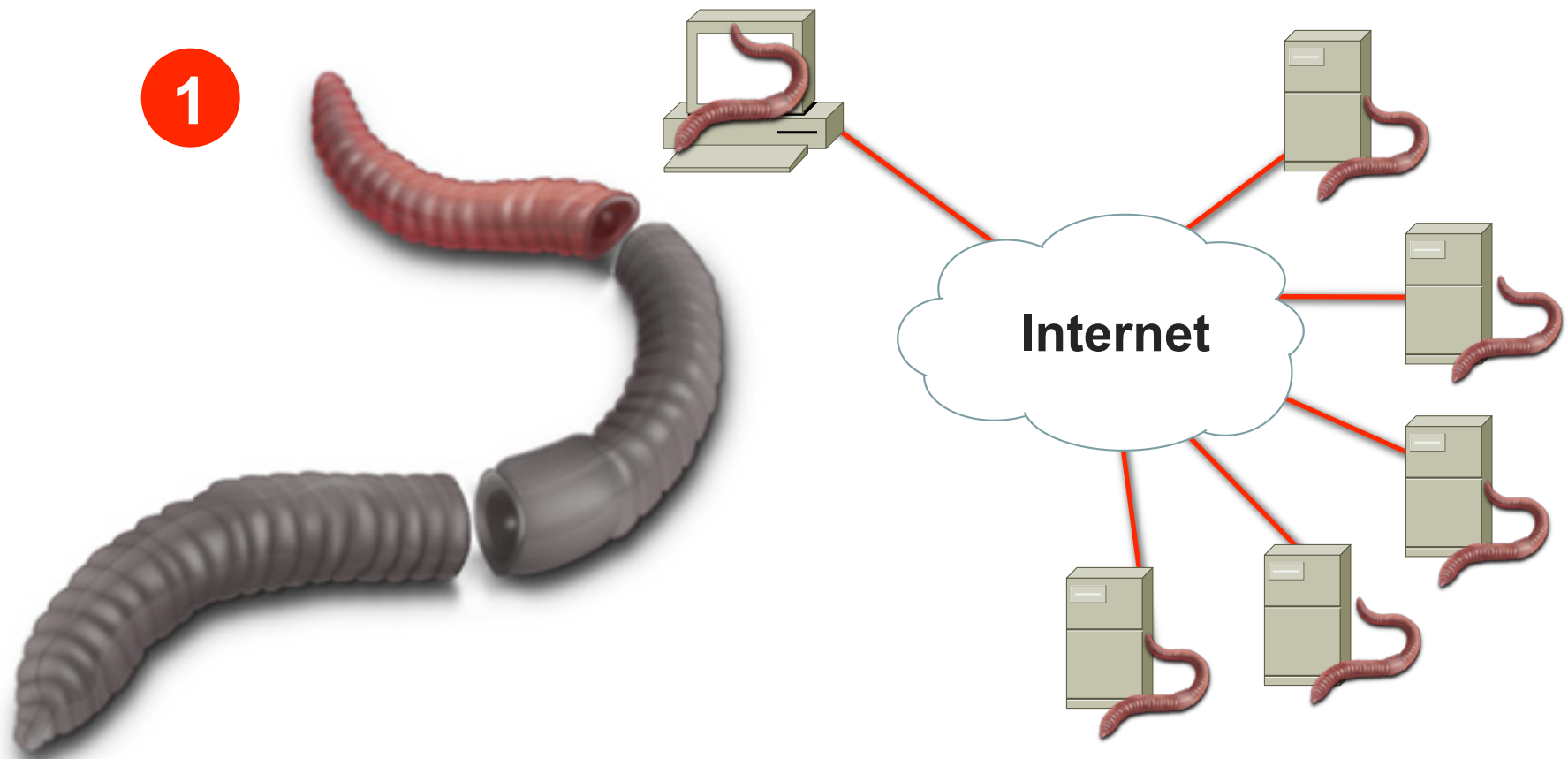
Anatomy of a Worm

**1—The Enabling
Vulnerability**

**2—Propagation
Mechanism**

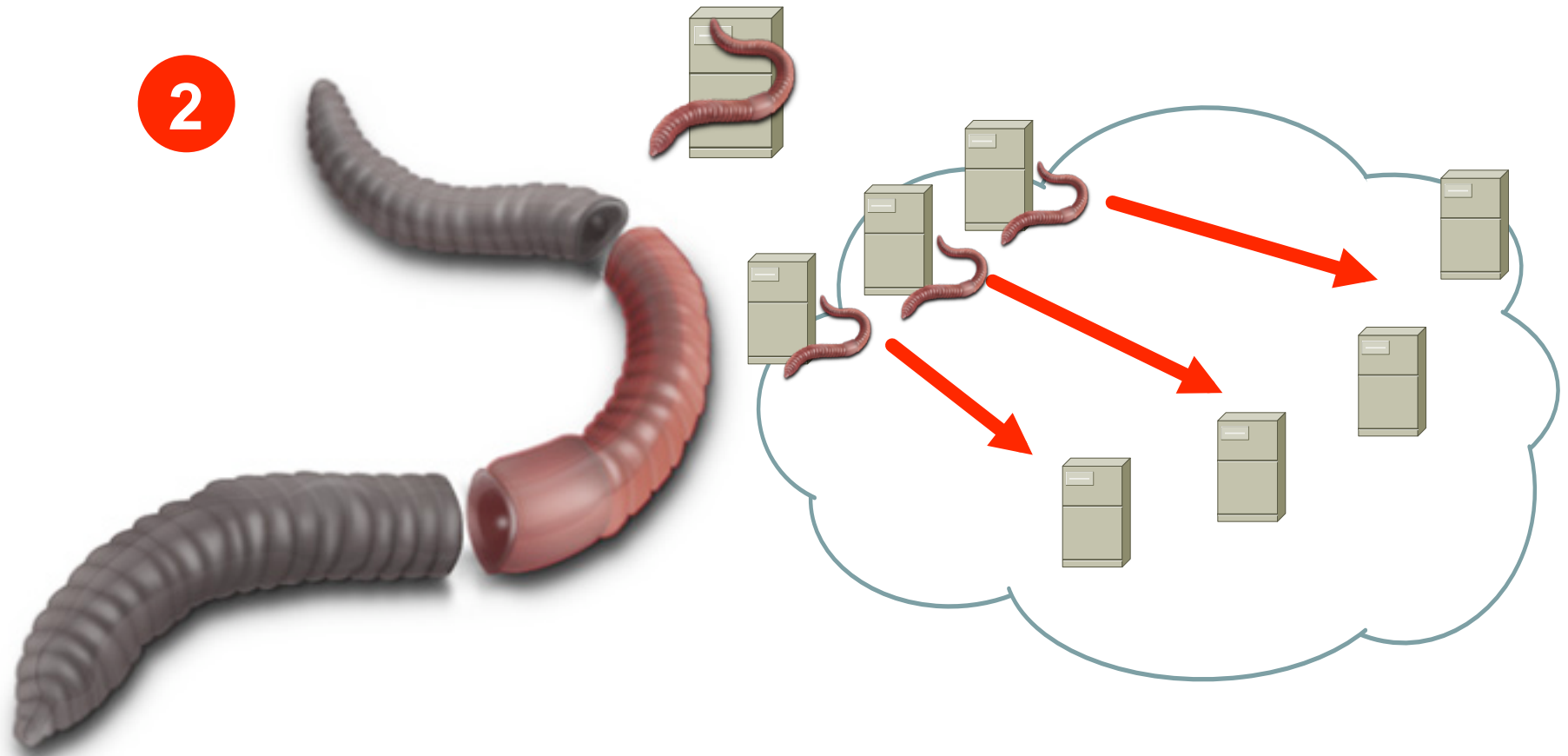
3—Payload

The Enabling Vulnerability



A Worm Installs Itself Using an Exploit Vector on a Vulnerable System

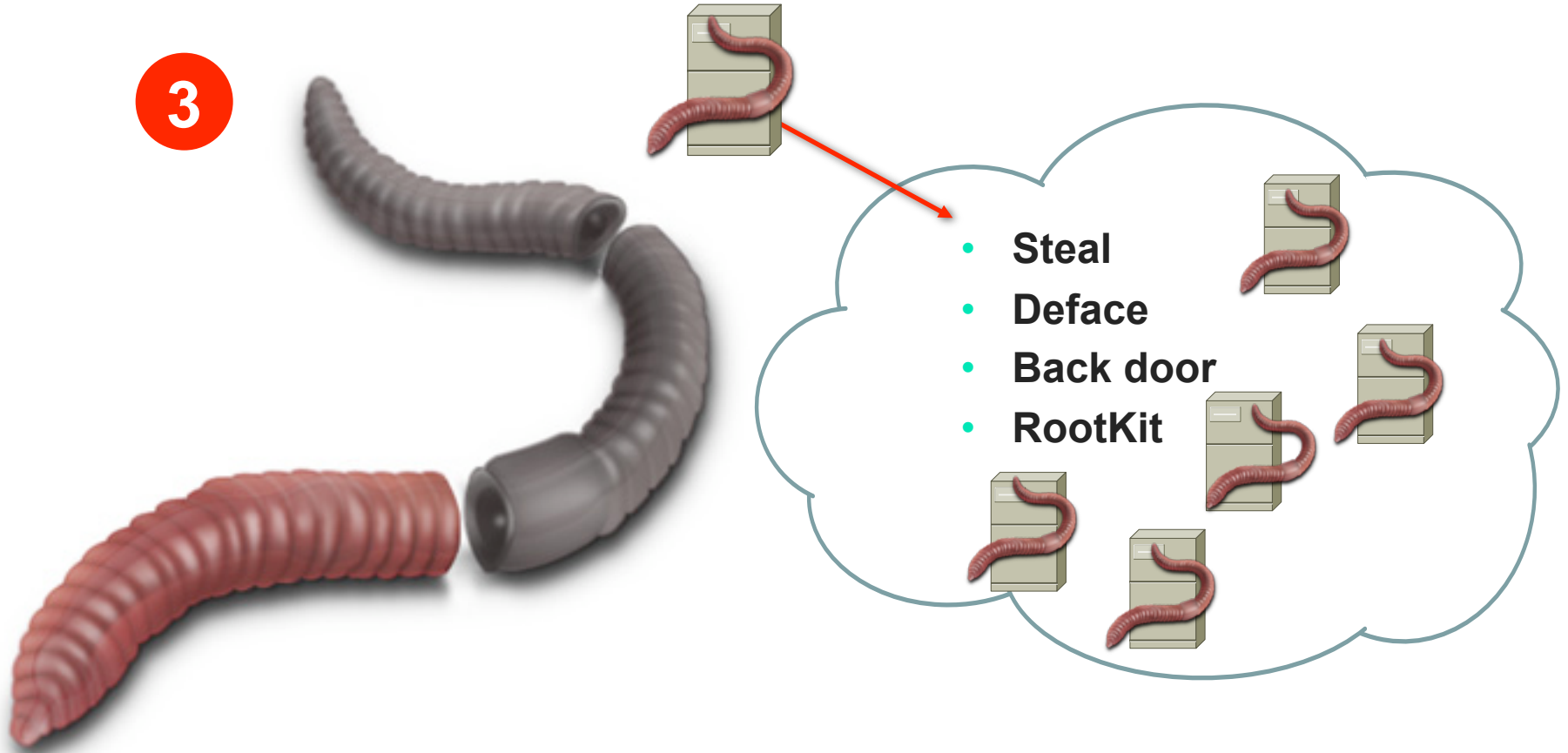
Propagation



**After Gaining Access to Devices,
Worm Replicates and Selects New Targets**

Payload

3





Worms and the Infrastructure

- Worms typically infect end-stations
- To date, worms have not targeted infrastructure BUT secondary effects have wreaked havoc
 - Increased traffic
 - Random scanning for destination
 - Destination address is multicast
 - TTL and other header variances
- At the core SP level, the aggregate affects of a worm can be substantial
- Worm severity is escalating and evolving

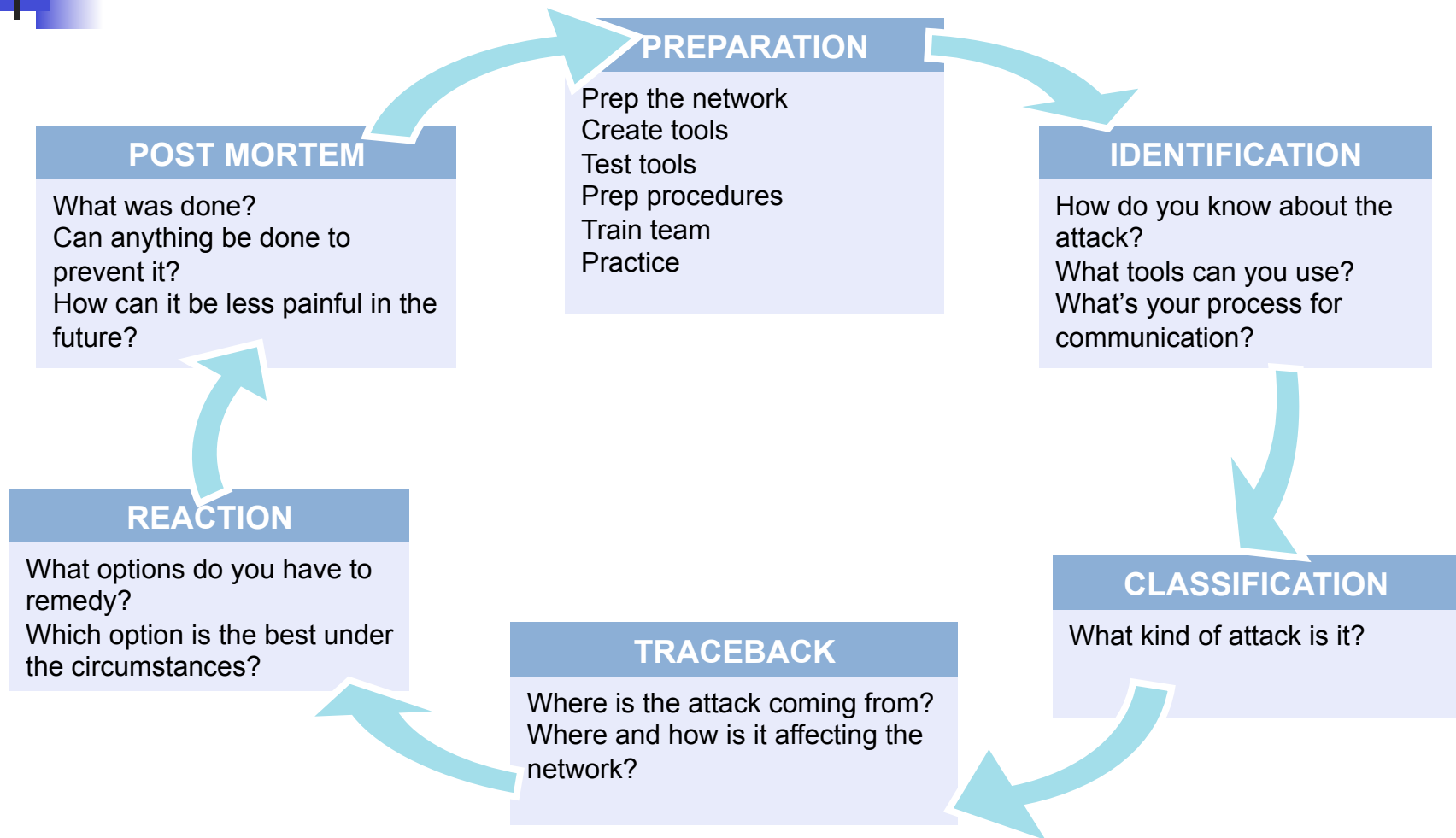


How Do You Respond?

With Money Being the Key Driver of Miscreant Activity, Large Network Operators Will Need to Respond

- BCP deployment
- Execution of a broad and deep security toolkit
- Rethink some network/service architectures
- Create, staff, and train an operational security (OPSEC) team
- Practice! Practice! Practice!

Six Phases of Incident Response





Preparation

Preparation—Develop and Deploy a Solid Security Foundation

- Includes technical and non-technical components
- Encompasses best practices
- The hardest, yet most important phase
- Without adequate preparation, you are destined to fail
- The midst of a large attack is not the time to be implementing foundational best practices and processes



Preparation

- Know the enemy
 - Understand what drives the miscreants
 - Understand their techniques
- Create the security team and plan
 - Who handles security during an event? Is it the security folks? The networking folks?
- Harden the devices
- Prepare the tools
 - Network telemetry
 - Reaction tools
 - Understand performance characteristics



Identification

Identification—How Do You Know You or Your Customer Is Under Attack?

- It is more than just waiting for your customers to scream or your network to crash
- What tools are available?
- What can you do today on a tight budget?



Ways to Detect

- Customer call
 - “The Internet is down”
- Unexplained changes in network baseline
 - SNMP: line/CPU overload, drops
 - Bandwidth
 - NetFlow
- ACLs with logging
- Backscatter
- Packet capture
- Network IDS
- Anomaly detection



Network Baselines

- NMS baselines
- Unexplained changes in link utilization
 - Worms can generate a lot of traffic, sudden changes in link utilization can indicate a worm
- Unexplained changes in CPU utilization
 - Worm scans can affect routers/switches resulting in increased CPU both process and interrupt switched
- Unexplained syslog entries
- These are examples
 - Changes don't always indicate a security event!
 - Need to know what's normal in order to identify abnormal behavior



Classification

- Classification—understand the details and scope of the attack
 - Identification is not sufficient; once an attack is identified, details matter
 - Guides subsequent actions
- Identification and classification are often simultaneous



Classification

- Qualify and quantify the attack without jeopardizing services availability (e.g., crashing a router):
 - What type of attack has been identified?
 - What's the effect of the attack on the victim(s)?
 - What next steps are required (if any)?
- At the very least:
 - Source and destination address
 - Protocol information
 - Port information



Traceback

- Traceback—what are the sources of the attack?
 - How to trace to network ingress points
 - Your Internet connection is NOT the only vector
 - Understand your topology!
- Traceback to network perimeter
 - NetFlow
 - Backscatter
 - Packet accounting
- Retain attack data
 - Use to correlate interdomain traceback
 - Required for prosecution
 - Deters future attacks
 - Clarify billing and other disputes
 - Post mortem analysis



Reaction

Reaction—Do Something to Counter the Attack

- Should you mitigate the attack?
 - Where? How?
- No reaction is a valid form of reaction in certain circumstances
- Reaction often entails more than just throwing an ACL onto a router



Post Mortem

Post Mortem—Analyze the Event

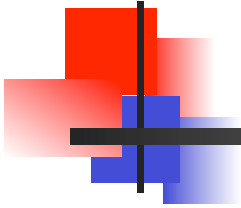
- The step everyone forgets!
- What worked? What didn't? How can we improve?
- What can be done to build build defense against repeat occurrences
- Was the DOS attack you just handled the real threat? Or was it a smoke screen for something else that just happened?
- What can you do to make it faster, easier, less painful in the future?
- Metrics are important!

■ Resources, headcount, etc.

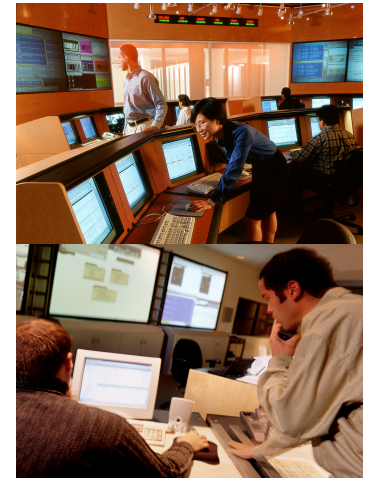
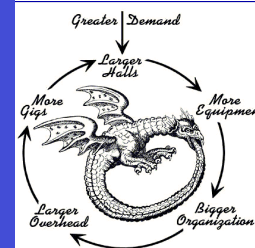
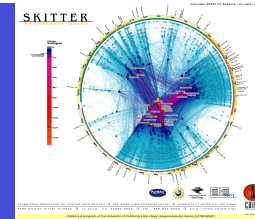


Security Workshop

OS Overview

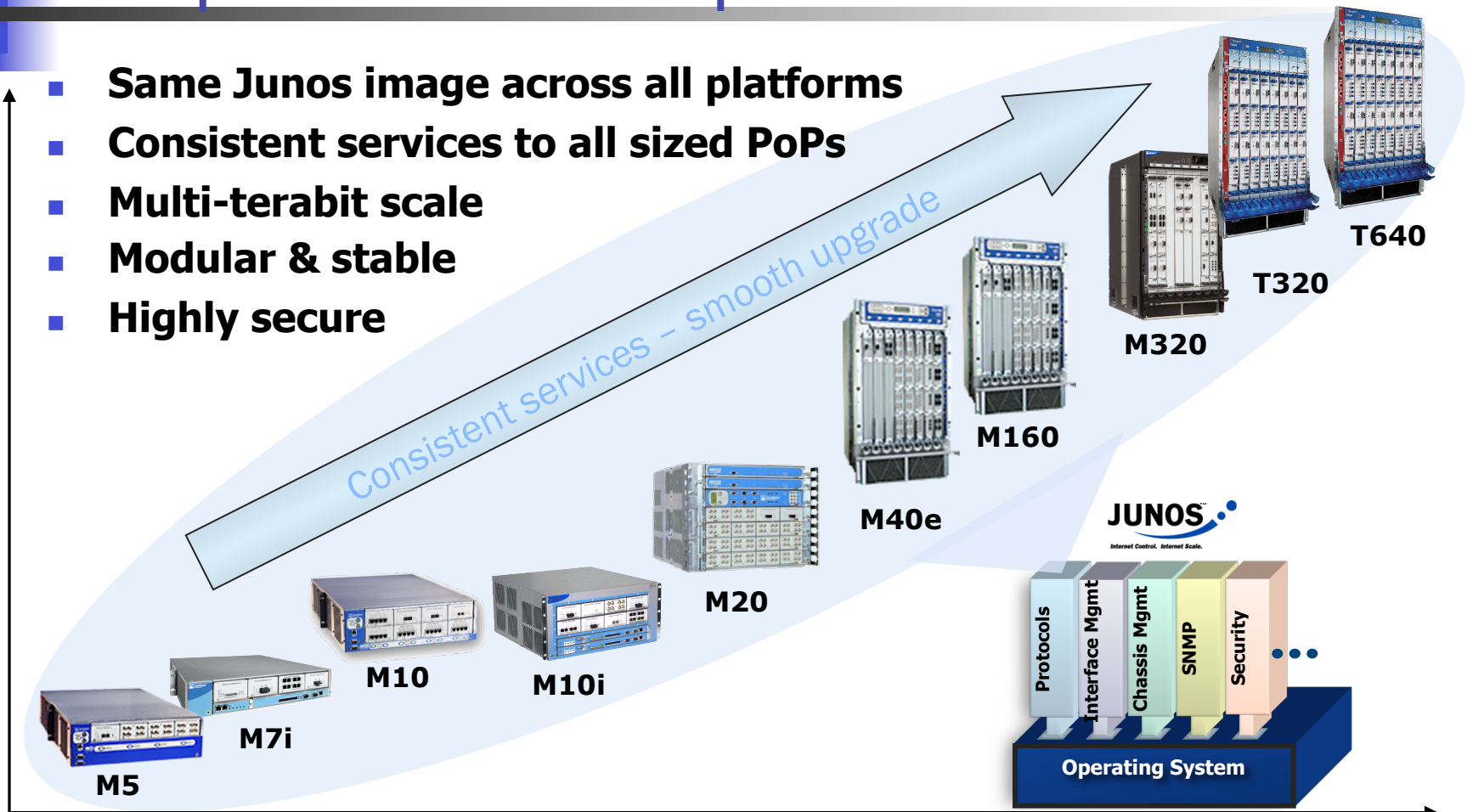


Juniper OS Configuration



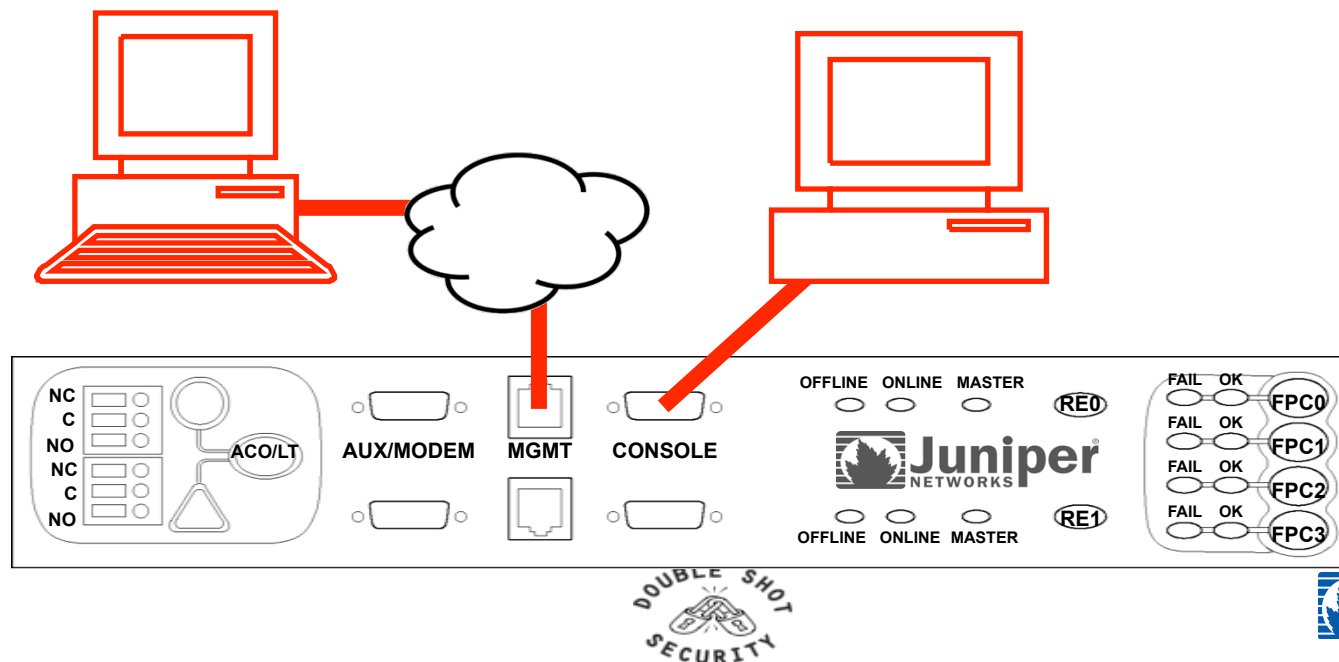
Juniper M-T series product line

- Same Junos image across all platforms
- Consistent services to all sized PoPs
- Multi-terabit scale
- Modular & stable
- Highly secure



Access Router's Management Ports

- Console
 - Db9 EIA-232 @ 9600 Bps, 8/N/1 (preconfigured)
- Management port, using Telnet, SSH
 - Requires configuration



Initial Login – JUNOS

- Log in as root

```
. . . .  
starting local daemons:..  
Fri Jan 17 22:23:32 UTC 1997
```

Amnesiac indicates a factory
default configuration

```
Amnesiac (ttyd0)
```

```
login: root  
Last login: Fri Jan 17 22:21:55 on ttyd0
```

```
--- JUNOS 5.2R2.3 built 2002-03-23 02:44:36 UTC
```

```
Terminal type? [vt100] <enter>  
root@%
```

BSD shell prompt

- Start CLI

```
root@% cli  
root>
```



Log In

- Router administrator configures login ID and password for each user
- Example session

```
lab2 (ttyd0)
```

```
login: perkins
```

```
Password:
```

```
Last login: Fri Feb 18 19:23:16 on ttyd0
```

```
Copyright (c) 1980, 1983, 1986, 1988, 1990, 1991, 1993, 1994
```

```
The Regents of the University of California.
```

```
---JUNOS 4.0R1 built 2000-02-10 09:29:44 UTC
```

```
perkins@lab2>
```




CLI Modes and Feature Overview

- CLI operational mode
 - Editing command lines
 - Command completion/history
 - Context-sensitive and documentation-based help
 - UNIX-style pipes
- CLI configuration mode
 - Object-oriented hierarchy
 - Configuration groups
 - Jumping between levels
 - Candidate configuration with sanity checking
 - Automatic rollback capability
 - Showing portions of configuration while configuring
 - Running operational-mode commands from within configuration
 - Saving, loading, and deleting configuration files
 - Wildcard deletes



CLI Modes

- Operational mode

- Monitor and troubleshoot the software, network connectivity, and router hardware

lab@host>

The > character identifies operational mode

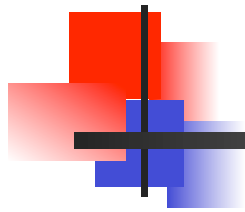
- Configuration mode

- Configure the router, including interfaces, general routing information, routing protocols, user access, and system hardware properties

[edit]

lab@host#

The # character identifies configuration mode



EMAC Style Shortcuts

Start of line	Ctrl-a
End Of line	Ctrl-e
Delete line	Ctrl-u, Ctrl-x
Delete cursor to end of line	Ctrl-k
Delete prev word	Ctrl-w
Redraw line	Ctrl-l
Search History	Ctrl-r Many more...



Command Completion

- Space bar completes a command

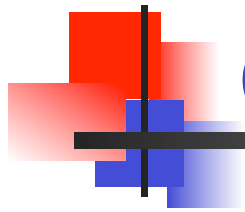
```
root@lab2> sh<space>ow i<space>  
'i' is ambiguous.
```

Possible completions:

igmp	Show information about IGMP
interfaces	Show interface information
isis	Show information about IS-
IS	

```
root@lab2> show i
```

- Tab key completes a variable



Context-Sensitive Help

Type a question mark (?) anywhere on command line

```
lab@host> ?
```

Possible completions:

clear	Clear information in the system
configure	Manipulate software configuration information
file	Perform file operations
help	Provide help information
. . .	

```
lab@host> clear ?
```

Possible completions:

arp	Clear address resolution information
bfd	Clear Bidirectional Forwarding Detection information
bgp	Clear Border Gateway Protocol information
cli	Clear command-line interface settings
firewall	Clear firewall counters
. . .	



Topical Help

The `help topic` command provides information on general concepts

```
lab@host> help topic icmp ?
```

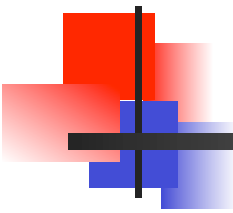
Possible completions:

<code>address</code>	IP addresses to include in router advertisements
<code>lifetime</code>	How long addresses in advertisements are valid
<code>min-advertisement-interval</code>	Time between router advertisements
<code>traceoptions</code>	Trace options for ICMP

```
lab@host> help topic icmp lifetime
```

Modify the Router Advertisement Lifetime

The `lifetime` field in router advertisement messages indicates how long a host should consider the advertised address to be valid. If this amount of time passes and the host has not received a router advertisement from the server, the route marks the advertised.....



Getting Help on Configuration Syntax

The `help reference` command provides configuration-related information

```
lab@host> help reference icmp lifetime
lifetime
    Syntax
```

```
lifetime seconds;
```

```
    Hierarchy Level
```

```
[edit protocols router-discovery interface interface-name]
```

```
    Description
```

```
How long the addresses sent by the server in its router advertisement
packets are valid. This time must be long enough so that another
. . . .
```

```
    Options
```

```
seconds--Lifetime value. A value of 0 indicates that one or more
addresses are no longer valid.
```

```
Range: 0, max-advertisement-interval value through 2 hours, 30
minutes (9000 seconds), specified in seconds
```

```
Default: 1800 seconds (30 minutes; three times the default
```



By the way...apropos

Where is keyword stub?

[edit]

```
lab@Hong_Kong_01# help apropos stub
```

...

[edited for brevity]

...

```
set protocols ospf sham-link no-advertise-local
set protocols ospf area <area_id>
set protocols ospf area <area_id> stub
set protocols ospf area <area_id> stub default-metric
    <default-metric>
set protocols ospf area <area_id> stub summaries
set protocols ospf area <area_id> nssa
```




Using | (Pipe)

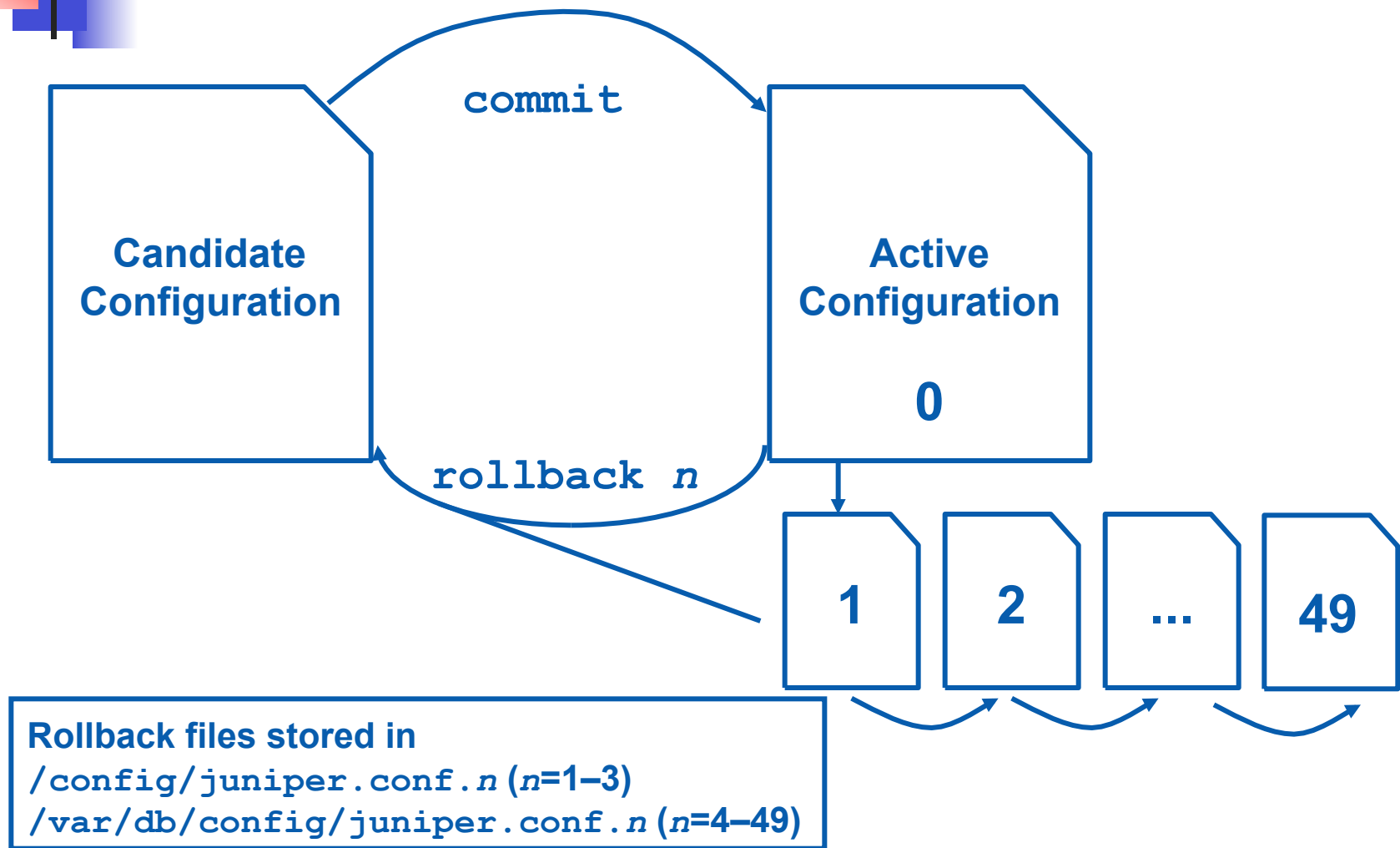
- The pipe function is used to filter output
 - Available in all modes and context

```
user@host> show route | ?
```

Possible completions:

count	Count occurrences
display	Display additional information
except	Show only text that does not match a pattern
find	Search for the first occurrence of a pattern
hold	Hold text without exiting the --More-- prompt
last	Display the last screen of lines in the output
match	Show only text that matches a pattern
no-more	Don't paginate output
request	Make system-level requests
resolve	Resolve IP addresses
save	Save output text to a file
trim	Trim specified number of columns from start of line

Activating a Configuration (1 of 2)





Entering Configuration Mode

- Type `configure` or `edit` at the CLI operational-mode prompt

```
root@lab2> configure
Entering configuration mode
[edit]
root@lab2#
```

- To allow a single user to edit the configuration, type `configure exclusive`
- `configure private` allows the user to edit a private copy of the candidate configuration
 - Multiple users can edit private candidate configurations simultaneously
 - At commit time, the user's private changes are merged back into the global configuration

Activating a Configuration ...

Commit

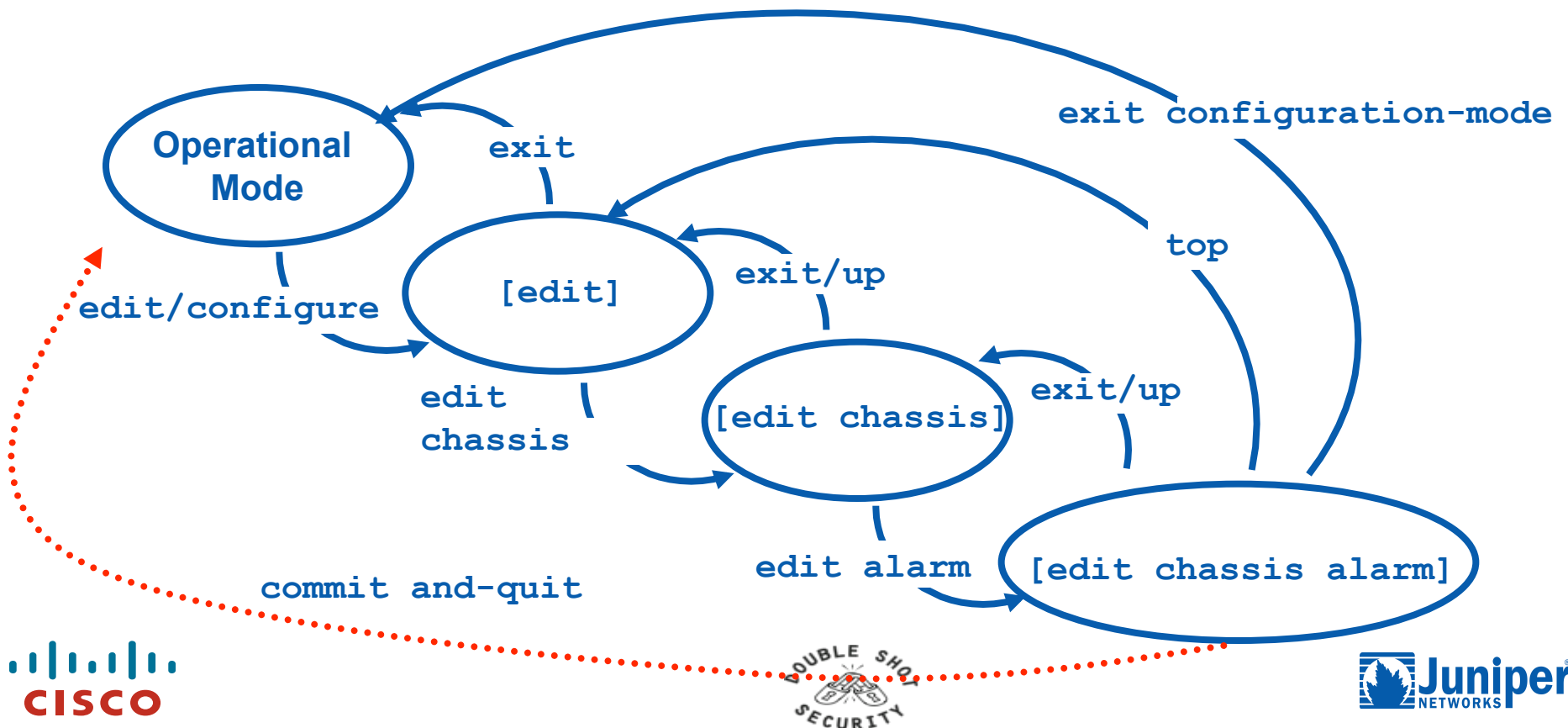
- Commit
 - Use `commit confirmed` to temporarily activate a configuration (default is 10 minutes). If configuration is not confirmed, router returns to previous configuration automatically; a second `commit` confirms the changes
- Use the `synchronize` switch to mirror the new configuration to a backup RE
- Support for scheduled and commented commits
 - Use the `commit at time` option (Release 5.5)

```
[edit]
user@host# commit at 20:01:00
configuration check succeeds
commit at will be executed at 2003-08-08 20:01:00 UTC
The configuration has been changed but not committed
Exiting configuration mode
```
 - Comments can be added to the `commits` log with the `comment` switch (Release 6.1)

Exiting Configuration Mode

Exiting levels

- Use `exit` from top level
- Use `exit configuration-mode` from any level
- Use `commit and-quit` as a time-saver





Configuration Hierarchy

- Create a hierarchy of configuration statements

- Enter commands in CLI configuration mode

```
root@lab2# set chassis alarm sonet lol red
```

- And the resulting configuration hierarchy is created...

```
chassis {  
  alarm {  
    sonet {  
      lol red;  
    }  
  }  
}
```

- Delete commands

```
root@lab2# delete chassis alarm sonet lol
```



Configuring Logical Interfaces

- Use the `set` command to configure a logical interface using the unit number

- For example:

```
lab@omaha> configure
```

```
[edit]
```

```
lab@omaha# set interfaces so-1/0/3 unit 40 dlci 40
```

- Or park yourself at the `unit` level:

```
lab@omaha> configure
```

```
[edit]
```

```
lab@omaha# edit interfaces so-1/0/3 unit 40
```

```
[edit interfaces so-1/0/3 unit 40]
```

```
lab@omaha# set dlci 40
```

Viewing Candidate Configuration

You can display just the portions that concern you from the root of the hierarchy...

```
[edit]
user@host# show chassis alarm
sonet {
    los red;
    pll yellow;
}
[edit]
```

```
user@host# edit chassis alarm
[edit chassis alarm]
user@host# show
sonet {
    los red;
    pll yellow;
}
[edit chassis alarm]
```

...or use edit to park yourself at a specific sub-hierarchy



run / do is Cool

- Use the run command to execute operational-mode CLI commands from within configuration
 - Can be a real time-saver when testing the effect of a recent change

```
[edit interfaces so-0/1/1]
lab@Amsterdam# set unit 0 family inet address 10.0.24.2/24
```

```
[edit interfaces so-0/1/1]
lab@Amsterdam# commit
commit complete
```

**Test configuration changes without
leaving configuration mode with run**

```
[edit interfaces so-0/1/1]
lab@Amsterdam# run ping 10.0.24.1 count 1
PING 10.0.24.1 (10.0.24.1): 56 data bytes
64 bytes from 10.0.24.1: icmp_seq=0 ttl=255 time=0.967 ms

--- 10.0.24.1 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.967/0.967/0.967/0.000 ms
```



Initial Configuration Checklist

- The following items are normally configured at initial system installation:
 - Root password
 - Host name
 - Domain name and DNS server address
 - Configuration file compression
 - System logging
 - Out-of-band management interface
 - Default and backup routers for management network
 - Configure system services for remote access
 - User accounts
 - System time
 - Loopback and transient interfaces
 - Remaining configuration needed to place the router into service (protocols, firewall filters, etc.)

Initial Configuration (1 of 10)

- Log in as root

```
. . . .  
starting local daemons:..  
Fri Jan 17 22:23:32 UTC 1997
```

Amnesiac indicates a factory
default configuration

```
Amnesiac (ttyd0)
```

```
login: root
```

```
Last login: Fri Jan 17 22:21:55 on ttyd0
```

```
--- JUNOS 5.2R2.3 built 2002-03-23 02:44:36 UTC
```

BSD shell prompt

```
Terminal type? [vt100] <enter>
```

```
root@%
```

- Start CLI

```
root@% cli
```

```
root>
```



Initial Configuration (2 of 10)

- Enter configuration mode

```
root> configure
[edit]
root#
```

- Configure root password

- Plain text

```
root# set system root-authentication plain-text-  
password
```

- Pre-encrypted password

```
root# set system root-authentication encrypted-  
password encrypted-password
```

**Do not enter a clear text
password in this mode!**



Initial Configuration - IOS(2a of 10)

- Enter configuration mode

```
Root# configure
```

```
Root(config)#
```

- Configure enable password

- Plain text

```
Root(config)# enable password password
```

- Pre-encrypted password

```
Root(config)# enable secret 5 $1!Q$hjHJHGJGJHGY
```

Do *not* enter a clear text password in this mode!



Initial Configuration (3 of 10)

- Configure router name

[edit]

```
root# set system host-name lab2
```

- Configure router domain name

[edit]

```
root# set system domain-name domain-name.tld
```

- Configure name server address

[edit]

```
root@# set system name-server ns-address
```



Initial Configuration – IOS (3a of 10)

- Configure router name

```
hostname lab2
```

- Configure router domain name

```
ip domain-name domain-name.tld
```

- Configure name server address

```
[edit]
```

```
ip name-server ns-address
```

Initial Configuration (4 of 10)

- Adjust syslog parameters as needed
 - Interactive command and configuration change logging is a good idea
 - Adjusting archive settings for more history also recommended

```
[edit system syslog]
root@lab2# show
user * {
    any emergency;
}
file messages {
    any notice;
    authorization info;
    archive size 1m files 20;
}
file cli-commands {
    interactive-commands any;
    archive size 1m files 10;
}
file config-changes {
    change-log info;
    archive size 1m files 10;
}
```

Archive settings adjusted
on default syslog file

Interactive commands and
configuration changes

Initial Configuration (5 of 10)

- Commit changes so far

```
[edit]  
root# commit  
commit complete
```

Note host name takes
effect after the commit

```
[edit]  
root@lab2#
```

- Configure management interface IP address and prefix

```
[edit]  
root@lab2# set interfaces fxp0 unit 0 family inet address ip-address/  
prefix-length
```

- Define a backup router

- Used when routing daemon is not running
 - Required when using redundant Routing Engines

```
[edit]  
root@lab2# set system backup-router gateway-address
```



Initial Configuration (6 of 10)

- Define static route for OoB management network

```
[edit]
```

```
root@lab2# edit routing-options
```

```
[edit routing-options]
```

```
root@lab2# set static route ip-address/prefix-length  
next-hop OoB-next-hop-address no-readvertise
```

- Configure system services for remote access

```
[edit]
```

```
root@lab2# set system services ssh
```

```
[edit]
```

```
root@lab2# set system services telnet
```

```
[edit]
```

```
root@lab2# set system services ftp
```



Initial Configuration – IOS (6a of 10)

- Define static route for OoB management network
- Static route

```
Lab2(config)# ip route destination mask next-hop ???
```

- Configure system services for remote access

```
line vty 0 4  
login  
password cisco
```



More on Banner

- Legal requirements may vary, but be explicit

“This system may be accessed by authorized persons only. Unauthorized access is forbidden and subject to criminal and civil penalties, as well as company disciplinary actions. By accessing this system you acknowledge that your actions will be monitored.”

Initial Configuration (7 of 10)

Configure user accounts

- Use predefined login classes, or create your own

```
[edit system login]
root@lab2# show
user dr-data {
  full-name "The Doctor 'O Data";
  uid 2003;
  class superuser;
  authentication {
    encrypted-password "$1$B78jkPLd$8VVjFv6D.ZQQev/5rstET0"; # SECRET-DATA
  }
}
```

The user ID is created automatically
when not explicitly configured

```
[edit system login]
root@lab2# show | display set
set system login user dr-data full-name "The Doctor 'O Data"
set system login user dr-data uid 2003
set system login user dr-data class superuser
set system login user dr-data authentication encrypted-password "$1$B78jkPLd$8VVjFv6D.ZQQev/5rstET0"
```

The commands used to create
the *dr-data* user account,
courtesy of display set



More on classes

- Define expected roles
 - One class per role
- Super-user
 - Use sparingly, even for yourself, as accidents happen when wielding enormous power!!! 😊
- View-only
 - Good for script engines
 - Limited NOC capability
- Newer OS's allow enormous granularity
 - Interface control vs. control plane (OSPF BGP etc)



Initial Configuration IOS (7a of 10)

- Configure user accounts
 - Use predefined privilege, or create your own

```
username lab privilege 15 password 7 09404F0B485744
```



Initial Configuration (8 of 10)

- Configure time zone and manually set the time of day

- Configure time zone:

[edit]

```
root@lab2# set system time-zone America/Los_Angeles
```

- Set date and time manually

```
root@lab2> set date ?
```

Possible completions:

<time>	New date and time (YYYYMMDDhhmm.ss)
--------	-------------------------------------

ntp	Set date/time using Network Time
-----	----------------------------------

Protocol servers

```
root@lab2> set date 200405141017.20
```

```
Fri May 14 10:17:20 PDT 2004
```

- Or, configure NTP



Initial Configuration- IOS(8a of 10)

- Configure time zone and manually set the time of day
 - Configure time zone:
`clock timezone timezone`
 - Set date and time manually
`clock set hh:mm:ss`
- Or, configure NTP



Initial Configuration (9 of 10)

Configure loopback and transient interfaces

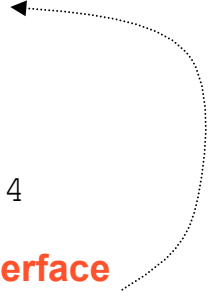
```
[edit interfaces]
root@lab2# set lo0 unit 0 family inet address 192.168.12.1

[edit interfaces]
root@lab2# set fe-0/0/2 unit 0 family inet address 10.0.13.2/24

[edit interfaces]
root@lab2# show lo0
unit 0 {
    family inet {
        address 192.168.12.1/32;
    }
}

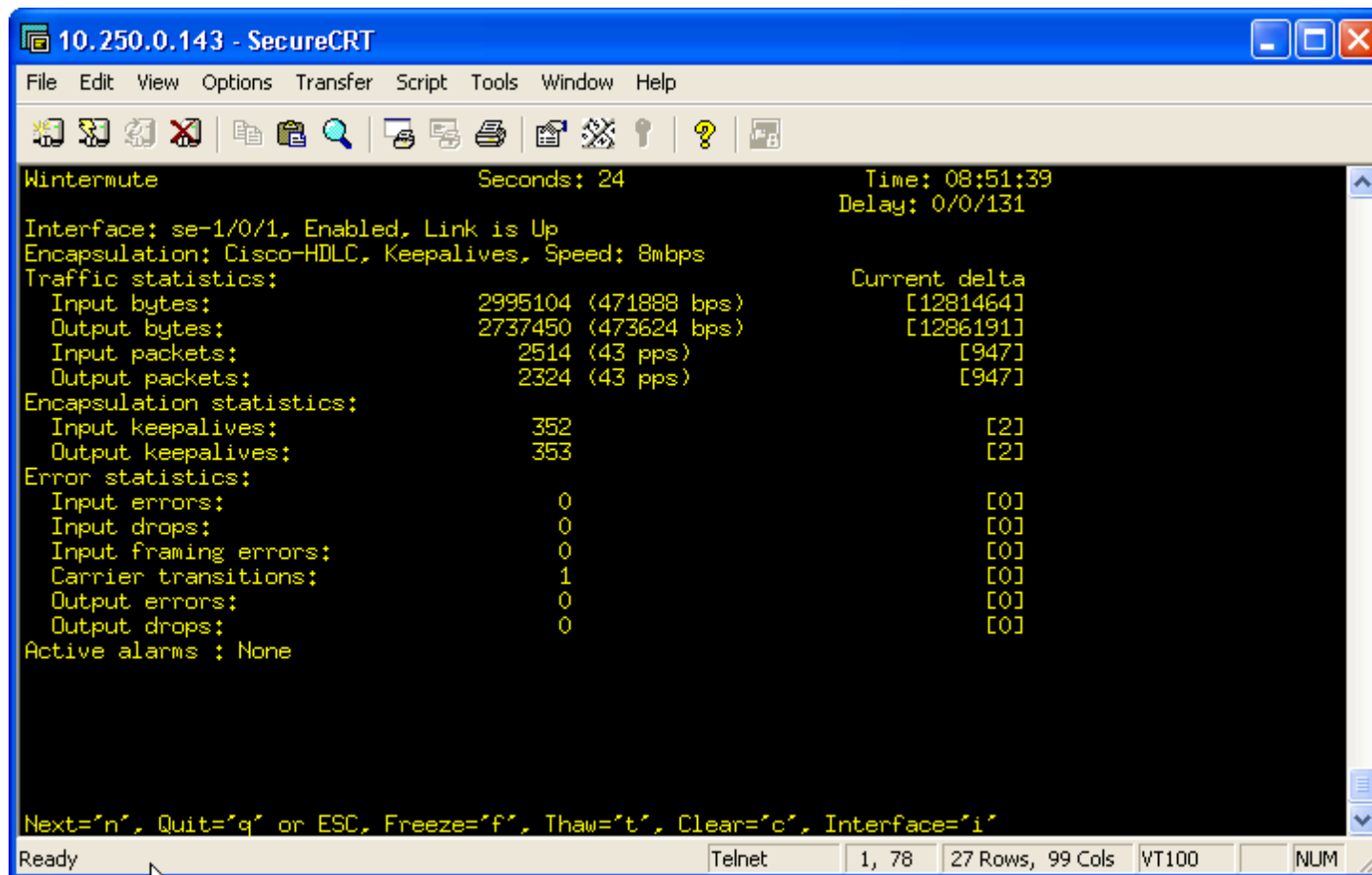
[edit interfaces]
root@lab2# show fe-0/0/2
unit 0 {
    family inet {
        address 10.0.13.2/24;
    }
}
```

Loopback interface
must use a /32



Monitoring an Interface

Use the `monitor interface` command for real-time statistics and error reports



The screenshot shows a SecureCRT terminal window titled "10.250.0.143 - SecureCRT". The terminal displays the output of the `monitor interface` command for interface `se-1/0/1`. The output includes traffic statistics, encapsulation statistics, error statistics, and active alarms. The terminal also shows a menu of commands at the bottom.

```
Wintermute                               Seconds: 24                               Time: 08:51:39
                                          Delay: 0/0/131

Interface: se-1/0/1, Enabled, Link is Up
Encapsulation: Cisco-HDLC, Keepalives, Speed: 8mbps
Traffic statistics:                               Current delta
Input bytes:                               2995104 (471888 bps)           [1281464]
Output bytes:                             2737450 (473624 bps)           [1286191]
Input packets:                             2514 (43 pps)             [947]
Output packets:                            2324 (43 pps)             [947]
Encapsulation statistics:
Input keepalives:                           352                       [2]
Output keepalives:                          353                       [2]
Error statistics:
Input errors:                               0                         [0]
Input drops:                               0                         [0]
Input framing errors:                       0                         [0]
Carrier transitions:                        1                         [0]
Output errors:                              0                         [0]
Output drops:                              0                         [0]
Active alarms : None

Next="n", Quit="q" or ESC, Freeze="f", Thaw="t", Clear="c", Interface="i"
```

Disabling, Deactivate, and Bounce

- Configuration mode deactivate and disable
 - `deactivate` causes the statement or hierarchy to be ignored
 - Comments out that portion of the configuration
 - `disable` administratively disables an interface or logical unit while retaining configured properties
- Use the operational-mode `request chassis` command to bounce FPC (PICs)
 - A warm boot of the FPC/PIC can clear problems
 - Less drastic than a chassis reboot and does not require configuration privileges

```
lab@Wintermute> request chassis fpc ?
```

Possible completions:

offline	Take FPC offline
online	Bring FPC online
restart	Restart FPC
slot	FPC slot number (0..6)

```
lab@Wintermute> request chassis fpc restart slot 2
```

Restart initiated, use "show chassis fpc" to verify

Network Utilities

■ Extended PING capabilities all on one line

```
lab@HongKong> ping ?
```

Possible completions:

<host>	Hostname or IP address of remote host
atm	Ping remote Asynchronous Transfer Mode node
bypass-routing	Bypass routing table, use specified interface
count	Number of ping requests to send (1..2000000000 packets)
detail	Display incoming interface of received packet
do-not-fragment	Don't fragment echo request packets (IPv4)
inet	Force ping to IPv4 destination
inet6	Force ping to IPv6 destination
interface	Source interface (multicast, all-ones, unrouted packets)
interval	Delay between ping requests (seconds)
logical-router	Name of logical router
+ loose-source	Intermediate loose source route entry (IPv4)
mpls	Ping label-switched path
no-resolve	Don't attempt to print addresses symbolically
pattern	Hexadecimal fill pattern
rapid	Send requests rapidly (default count of 5)
record-route	Record and report packet's path (IPv4)
[deleted for brevity]	



Access command options by
clicking **Advanced options** on the Juniper®
NETWORKS logo

Network Utilities (Cont.)

- The `monitor traffic` command provides CLI access to the `tcpdump` utility
 - Only displays traffic originating or terminating on local RE
 - The best way to perform analysis of Layer 2 protocols in JUNOS software
 - Protocol filtering currently requires writing and reading from a file (hidden `write-file` and `read-file` options)

```
lab@host> monitor traffic interface ge-0/3/0 detail
```

```
Listening on ge-0/3/0, capture size 96 bytes
```

```
16:20:24.043006 In IP (tos 0x0, ttl 255, id 53152, offset 0, flags [none],  
length: 84) 10.0.16.1 > 10.0.16.2: icmp 64: echo request
```

```
16:20:24.043061 Out IP (tos 0x0, ttl 255, id 57238, offset 0, flags [none],  
length: 84) 10.0.16.2 > 10.0.16.1: icmp 64: echo reply
```

```
. . .
```

ICMP echo
traffic

Tracing Overview

- Tracing is the JUNOS software equivalent of *debug*
 - Can be enabled on a production network
 - Requires configuration
 - Multiple options (flags) can be traced to a single file
- Generic tracing configuration syntax

The protocol/function being traced

Where to write the trace results

```
[edit protocols protocol-name]
```

```
user@host# show
```

```
  traceoptions {
```

```
    file filename [replace] [size size] [files number] [no-stamp];
```

```
    flag flag [flag-modifier] [disable];
```

```
  }
```

Flags identify what aspects of the protocol is traced and at what level of detail



Protocol Tracing

- Include the `traceoptions` statement at the `[edit protocols protocol-name]` hierarchy
 - Useful when troubleshooting configuration and interoperability problems
- A typical BGP tracing configuration is shown along with sample output:

```
[edit protocols bgp]
```

```
lab@host# show
```

```
traceoptions {
```

```
    file bgp-trace;
```

```
    flag open detail;
```

```
    flag update detail;
```

```
    flag keepalive detail;
```

```
}
```

```
lab@host> show log bgp-trace
```

```
. . .
```

```
Feb 19 16:07:47 BGP RECV 192.168.2.1+2705 -> 192.168.0.1+179
```

```
Feb 19 16:07:47 BGP RECV message type 1 (Open) length 45
```

```
Feb 19 16:07:47 BGP RECV version 4 as 10 holdtime 90 id 192.168.2.1 parmlen 16
```

```
Feb 19 16:07:47 BGP RECV MP capability AFI=1, SAFI=1
```

```
. . .
```




Analyzing Log and Trace Files

- Use the `show log file-name` CLI command to display contents of log and tracefiles
 - Hint: Get help on available options at the `more` prompt by entering an `h`
- Do not forget the CLI's pipe functionality; it makes log parsing a breeze!
 - Cascade pipe instances to evoke a logical AND search; use quotes to evoke a logical OR, as shown:

```
lab@host> show log messages | match fail
```

```
Jan 29 12:40:47  Montreal-3  rpd[2228]: RPD_ISIS_ADJDOWN: IS-IS lost  
L2 adjacency to Amsterdam-3 on so-0/3/1.0, reason: 3-Way Handshake  
Failed
```

```
lab@London> show log messages | match "fpc | error | kernel | panic"
```



Miscellaneous Log File Commands

- Monitor a log/trace in real time with the CLI's `monitor` command

```
user@host> monitor start filename
```

- Shows updates to monitored file(s) until canceled, with piped output matching!
- Use Esc-Q to enable/disable real-time output to screen
- Issue `monitor stop` to cease all monitoring

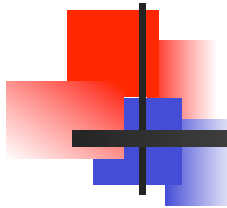
- Log/trace file manipulation

- Use the `clear` command to truncate (clear) log/trace files

```
user@host> clear log filename
```

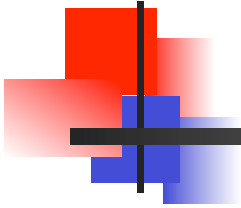
- Use the `file delete` command to delete log/trace files

```
user@host> file delete filename
```

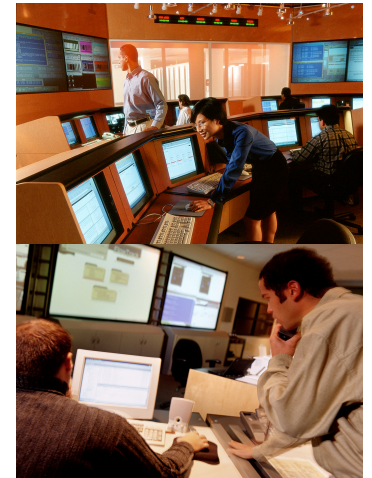
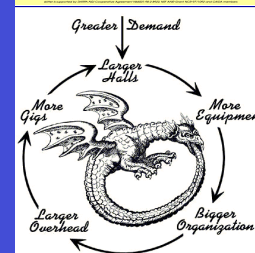
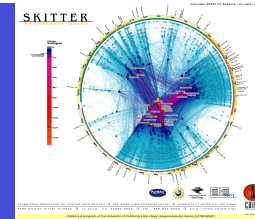


IOS <-> JUNOS

Basic CLI and Systems Management Commands	clock set	set date
	ping	ping
	reload	request system reboot
	send	request message
	show clock	show system uptime
	show environment	show chassis environment
	show history	show cli history
	show ip traffic	show system statistics
	show logging	show log show log file name
	Show processes	show system processes
	show running config	show configuration
	show tech-support	request support information
	show users	show system users
	show version	show version show chassis hardware
	terminal length	set cli screen-length
	terminal width	set cli screen-width
	trace	traceroute



Cisco IOS Configuration





Router Components

- Bootstrap – stored in ROM microcode – brings router up during initialisation, boots router and loads the IOS.
- POST – Power On Self Test - stored in ROM microcode – checks for basic functionality of router hardware and determines which interfaces are present
- ROM Monitor – stored in ROM microcode – used for manufacturing, testing and troubleshooting
- Mini-IOS – a.k.a RXBOOT/boot loader by Cisco – small IOS ROM used to bring up an interface and load a Cisco IOS into flash memory from a TFTP server; can also do a few other maintenance operations



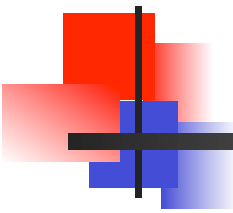
Router Components

- RAM – holds packet buffers, ARP cache, routing table, software and data structure that allows the router to function; running-config is stored in RAM, as well as the decompressed IOS in later router models
- ROM – starts and maintains the router
- Flash memory – holds the IOS; is not erased when the router is reloaded; is an EEPROM [Electrically Erasable Programmable Read-Only Memory] created by Intel, that can be erased and reprogrammed repeatedly through an application of higher than normal electric voltage
- NVRAM – Non-Volatile RAM - holds router configuration; is not erased when router is reloaded



Router Components

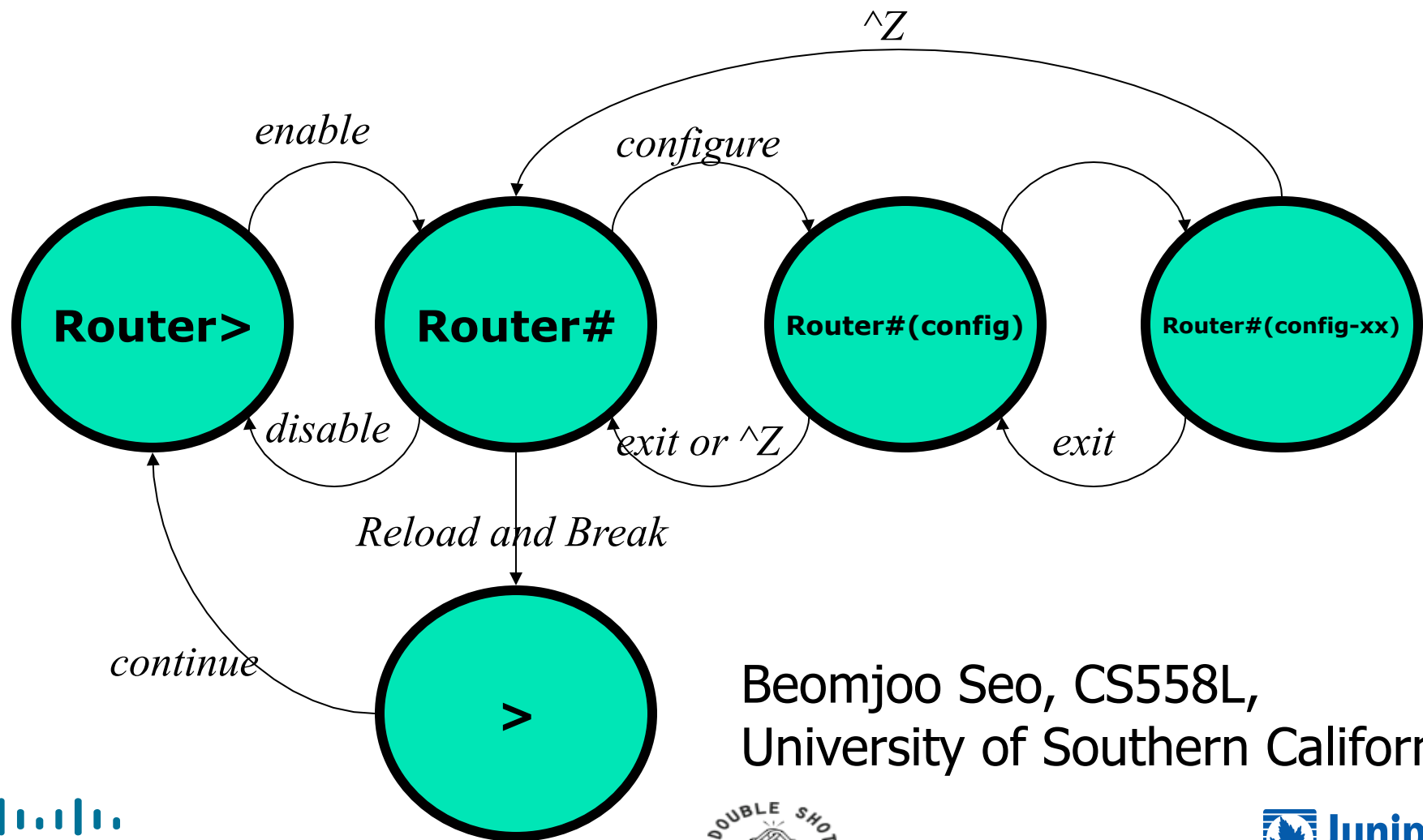
- Config-Register – controls how router boots; value can be seen with “`show version`” command; is typically 0x2102, which tells the router to load the IOS from flash memory and the `startup-config` file from NVRAM



Router Modes Changed With Config-Register

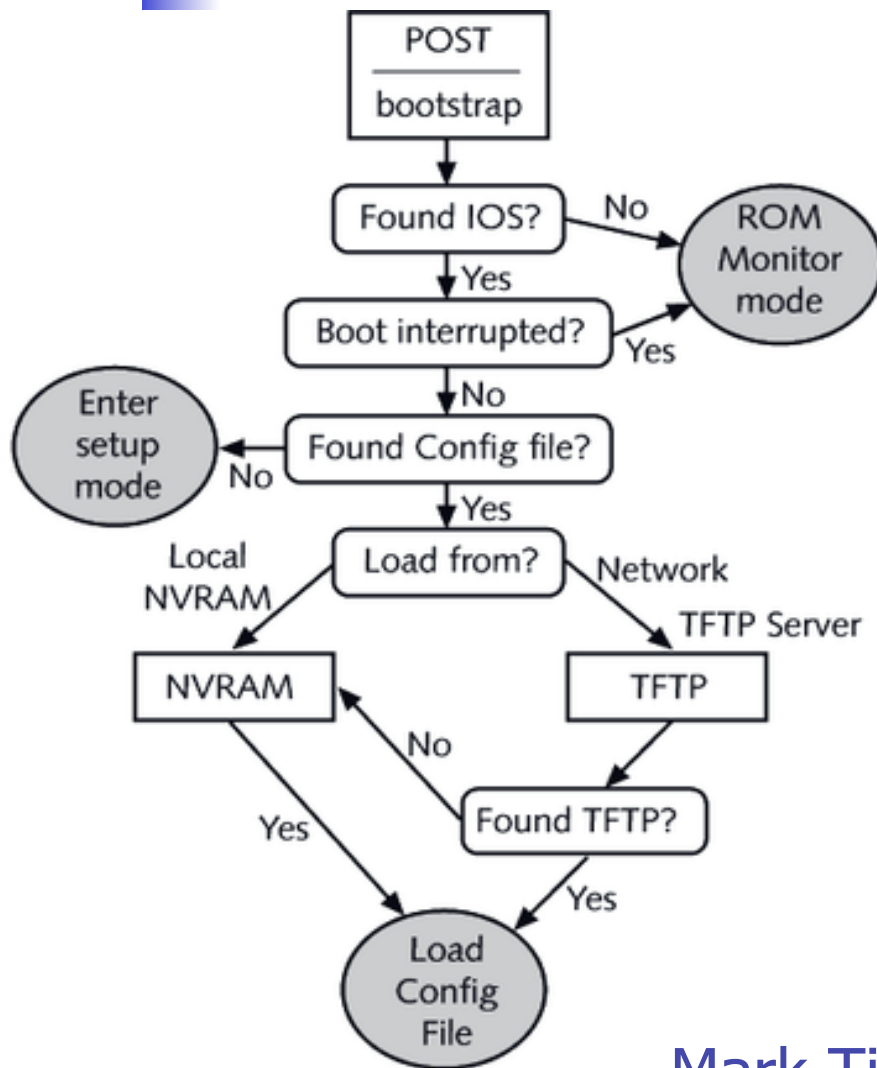
- Reasons why you would want to modify the config-register:
 - Force the router into ROM Monitor Mode
 - Select a boot source and default boot filename
 - Enable/Disable the Break function
 - Control broadcast addresses
 - Set console terminal baud rate
 - Load operating software from ROM
 - Enable booting from a TFTP server

Router Modes Change and Prompts



Beomjoo Seo, CS558L,
University of Southern California

Router Setup and Startup



POST – loaded from ROM and runs diagnostics on hardware
Bootstrap – locates and loads the IOS image; default is flash
IOS – locates and loads a valid configuration from NVRAM; from startup-config and only exists if you copy running-config to NVRAM

Startup-config – if found, router loads it and runs embedded configuration; if not found, router enters setup mode



Where is the Cisco IOS Configuration?

Command from Enable Mode	Description
copy running-config tftp	Copies the running configuration located in RAM to a TFTP server.
copy startup-config tftp	Copies the startup configuration located in NVRAM to a TFTP server.
copy tftp running-config	Copies the configuration from the TFTP server to the running configuration. The reconfiguration of the router is immediate when this command is issued. The running-config is not replaced. The files are blended.
copy tftp startup-config	Copies the configuration from the TFTP server to the startup configuration. The startup-config is replaced with the one from the TFTP server.
copy run start	Copies the working configuration file in RAM to the startup configuration file in NVRAM. Replaces the startup configuration file.
copy start run	Copies the startup configuration file in NVRAM to the running configuration in RAM. Does not replace the file in RAM; the files are blended.
copy flash tftp	Copies the IOS in flash memory to a TFTP server.
copy tftp flash	Copies the IOS from a TFTP server to flash memory.
configure terminal	Used to specify that you would like to configure your settings manually from the console terminal.
configure memory	Used to specify that you would like to pull your configuration information from NVRAM.



Router Access Modes

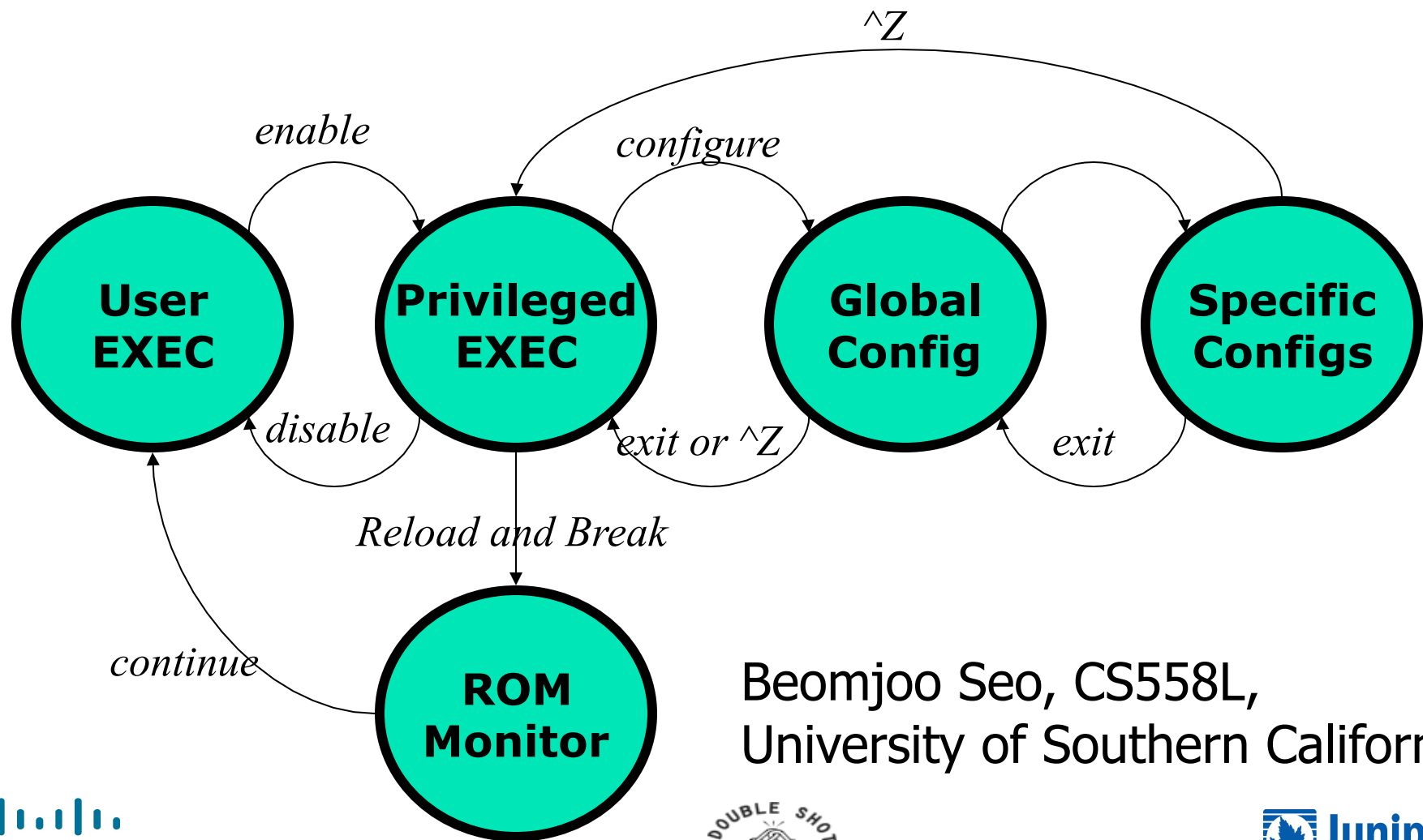
- User EXEC mode - limited examination of router
 - Router>
- Privileged EXEC mode - detailed examination of router, debugging, testing, file manipulation
 - Router#
- ROM Monitor - useful for password recovery & new IOS upload session
- Setup Mode – available when router has no `startup-config` file



Router Access Modes

Mode	Prompt	To enter	To exit	Used for
User EXEC	Router>	If there is a line password, enter it. Otherwise, press the Return or Enter key.	Logout or Exit	Shows the status of the router and allows network operators to manage connections
Privileged EXEC	Router#	Type enable at the prompt.	Disable Exit Logout	Copies, erases, sets up, and shows router settings
Global configuration	Router (config)#	Configure	Exit End	Allows you to configure various items, including clock, hostname, enable password, and enable secret password
Interface configuration	Router (config-if)#	Interface Ethernet0 or Interface Serial0	Exit End	Allows you to configure the settings, such as IP, for a specific interface
Line configuration	Router (config-line)#	Line console 0 or Line vty 0 4 or Line aux 0	Exit End	Configures lines, such as the console, virtual terminal, or auxiliary
Router configuration	Router (config-router)#	Router rip or Router igrp	Exit End	Adds or configures RIP, IGRP, or other routing protocols

CLI Modes for Router Access

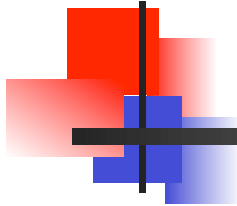


Beomjoo Seo, CS558L,
University of Southern California



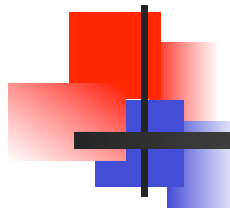
External Configuration Sources

- Console – direct PC serial access
- Auxilliary port – Modem access
- Virtual terminals – Telnet access
- TFTP Server – copy configuration file into router RAM
- Network Management Software - CiscoWorks



Telnet

- Utility that connects at the highest layer of the OSI model
- Provides remote access to other devices
- Cisco routers allow telnet connections via their virtual terminal ports
- If you can establish telnet connectivity to a router, you have established that it is available on the network and that you have connectivity at all layers



SSH

- Replaces telnet for a protected command and control communication channel
- Privacy and integrity provided through the use of strong cryptographic algorithms
- Supports TACACS+, RADIUS and local authentication
- Secure Copy (SCP) available in new SSH enabled code
- Restrict access to ssh via “transport input ssh” command
- SSHv2 now in Cisco IOS!



IP Host Names

- When telnetting to a remote router or host, the IP address of the host must follow the telnet command
- Rather than using IP addresses, it is easier to refer to a remote host or router using a name
- Sometimes, you cannot gain connectivity because the host name that you are trying to connect with is entered in a table incorrectly
- Using a name server provides name resolution from one location, making a table configuration on each device unnecessary



Ping and Trace

- If you can't get connectivity at the Application layer, try connectivity at the Internetwork layer
- Ping and trace verify connectivity at the Internetwork layer
 - Both use ICMP messages to verify the destination host is reachable, and if not, give possible reasons for the problem
- Ping sends a packet to the destination and waits for a response
 - By default, the ping utility with Cisco routers is configured to send five packets to the target



Ping and Trace

- Extended mode ping
 - Options include:
 - The destination address of the ping
 - The protocol
 - Repeat count
 - Datagram size
 - Can only be accessed from the privileged mode prompt



Ping and Trace

- If ping indicates a problem with connectivity, using trace may provide a better clue as to the source of the connectivity problem
- Trace command is similar to ping command, except that the replies are requested at each hop along the way to the destination
- Trace sends multiple ICMP packets with progressively higher TTL counters until the packet reaches the destination



IP Route

- If you cannot get connectivity using ping or trace, you should check your routing table
- You can issue the show ip route command from the enable mode prompt
 - This command shows the routing table
- Typically, routing tables are dynamically created when routing protocols are configured on the router



Checking the Interface

- On of the biggest mistakes made when troubleshooting is not checking the interfaces on the router
- If the interfaces are down, packets cannot be delivered
- Router interfaces go down for a variety of reasons including:
 - Incorrect IP configuration
 - Cable problems



Checking the Interface

- Keepalive frames
 - Data frames sent between two hosts to ensure that the connection between those hosts remains open
- Different types of interfaces can show different types of reports
 - For example, a Token Ring interface reports down when there is no electrical carrier signal present

Checking the Interface

```
lab-a#show interfaces
```

Ethernet0 is up, line protocol is up ←

Hardware is Lance, address is 0000.0c8e.b490 (bia 0000.0c8e.b490)
Internet address is 192.5.5.1/24
MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA, loopback not set
Keepalive set (10 sec)
ARP type: ARPA, ARP Timeout 04:00:00
Last input never, output 00:00:03, output hang never
Last clearing of "show interface" counters never
Queueing strategy: fifo
Output queue 0/40, 0 drops; input queue 0/75, 0 drops
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec

Serial0 is down, line protocol is down ←

Hardware is HD64570
Internet address is 201.100.11.1/24
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation HDLC, loopback not set
Keepalive set (10 sec)
Last input never, output never, output hang never
Last clearing of "show interface" counters never
Queueing strategy: fifo
Output queue 0/40, 0 drops; input queue 0/75, 0 drops
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts, 0 runs, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
15 packets output, 3198 bytes, 0 underruns
0 output errors, 0 collisions, 2 interface resets
0 output buffer failures, 0 output buffers swapped out
0 carrier transitions
DCD=up DSR=up DTR=down RTS=down CTS=up

Serial1 is administratively down, line protocol is down ←

Hardware is HD64570
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation HDLC, loopback not set
Keepalive set (10 sec)
Last input never, output never, output hang never

Interface E0 is fully functional. Frames can be sent and received on this interface.

S0 is not functional. In this case, the serial interface on the router attached to this router is down. If one end of a point-to-point link is down, it will "push" the attached up interface on the next router down.

The S1 interface is not functional. In this case, there is no cable attached to S1 as it is not being used.



Clear Counters

- Routers keep detailed statistics regarding the data passing across its interfaces
- Before using the show interface command, you may want to clear the existing interface information
- You can clear these statistics (**counters**) on the interface by using the clear interface or clear counters command



Debug

- Debug command
 - One of the most powerful tools you can use to obtain information from your router
 - Only available from privileged EXEC mode
 - Has numerous subcommands that allow you to troubleshoot particular protocols
 - Allows you to check for specific types of traffic on the wire

Debug

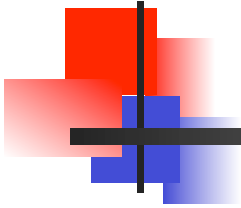
```
RouterB#debug all
This may severely impact network performance. Continue? [confirm]

All possible debugging has been turned on
RouterB#
IP: s=172.22.3.1 (Serial1), d=255.255.255.255, len 76, rcvd 2
UDP: rcvd src=172.22.3.1(520), dst=255.255.255.255(520), length=52
RIP: received v1 update from 172.22.3.1 on Serial1
      172.22.4.0 in 1 hops
      172.22.5.0 in 2 hops
RIP: Update contains 2 routes
SERVICE_MODULE(0): lxt441 interrupt 1 status A7 loop 0
SERVICE_MODULE(0): lxt441 interrupt 1 status 87 loop 0
SERVICE_MODULE(0): lxt441 interrupt 1 status A7 loop 0
SERVICE_MODULE(0): lxt441 interrupt 1 status 87 loop 0
Serial1: HDLC myseq 6631, mineseen 6631, yourseen 6580, line up
SERVICE_MODULE(0): lxt441 interrupt 1 status A7 loop 0
SERVICE_MODULE(0): lxt441 interrupt 1 status 87 loop 0
SERVICE_MODULE(0): lxt441 interrupt 1 status A7 loop 0
SERVICE_MODULE(0): lxt441 interrupt 1 status 87 loop 0
RIP: sending v1 update to 255.255.255.255 via Ethernet0 (172.22.2.1)
      subnet 172.22.3.0, metric 1
      subnet 172.22.4.0, metric 2
      subnet 172.22.5.0, metric 3
RIP: Update contains 3 routes
IP: s=172.22.2.1 (local), d=255.255.255.255 (Ethernet0), len 55, sending broad/m
ulticast
RIP: sending v1 update to 255.255.255.255 via Serial1 (172.22.3.2)
      subnet 172.22.2.0, metric 1
RIP: Update contains 1 routes
IP: s=172.22.3.2 (local), d=255.255.255.255 (Serial1), len 67, sending broad/mul
ticast
SERVICE_MODULE(0): lxt441 interrupt 1 status A7 loop 0
SERVICE_MODULE(0): lxt441 interrupt 1 status 87 loop 0
Serial0: attempting to restart
Serial1: HDLC myseq 6632, mineseen 6632, yourseen 6581, line up
IP: s=172.22.5.1 (Ethernet0), d=255.255.255.255, len 106, rcvd 2
UDP: rcvd src=172.22.5.1(520), dst=255.255.255.255(520), length=72
RIP: ignored v1 update from bad source 172.22.5.1 on Ethernet0
SERVICE_MODULE(0): lxt441 interrupt 1 status A7 loop 0
SERVICE_MODULE(0): lxt441 interrupt 1 status 87 loop 0
SERVICE_MODULE(0): lxt441 interrupt 1 status A7 loop 0
SERVICE_MODULE(0): lxt441 interrupt 1 status 87 loop 0
SERVICE_MODULE(0): lxt441 interrupt 1 status A7 loop 0
SERVICE_MODULE(0): lxt441 interrupt 1 status 87 loop 0
Serial1: HDLC myseq 6633, mineseen 6633, yourseen 6582, line up
All possible debugging has been turned off
RouterB#
```

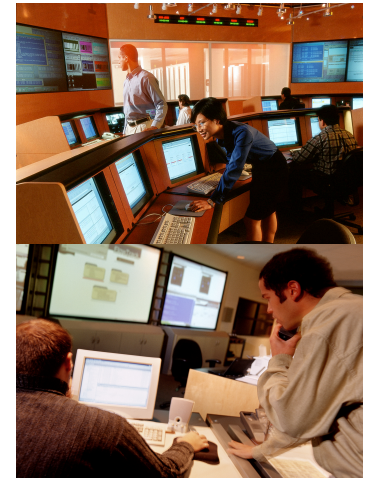
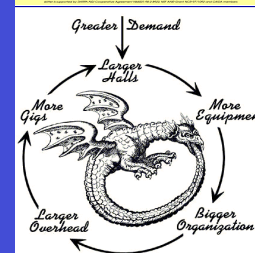
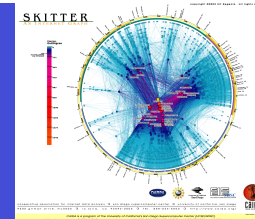
The debug all command warns you that issuing this command could cause severe network congestion. This command should only be used for a short period of time as a troubleshooting tool.



- Follow the lab-guide to set up initial topology



Infrastructure Security

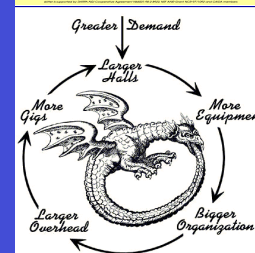
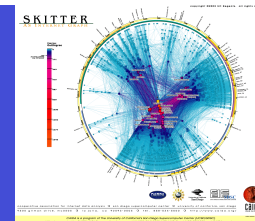




Infrastructure Security

- Best Common Practice [BCP]
 - Secure Router Access
 - Edge Protection
 - Remote triggered black hole filtering
 - Sink holes
 - Source address validation on all customer traffic
 - Control Plane Protection

Secure Router Access



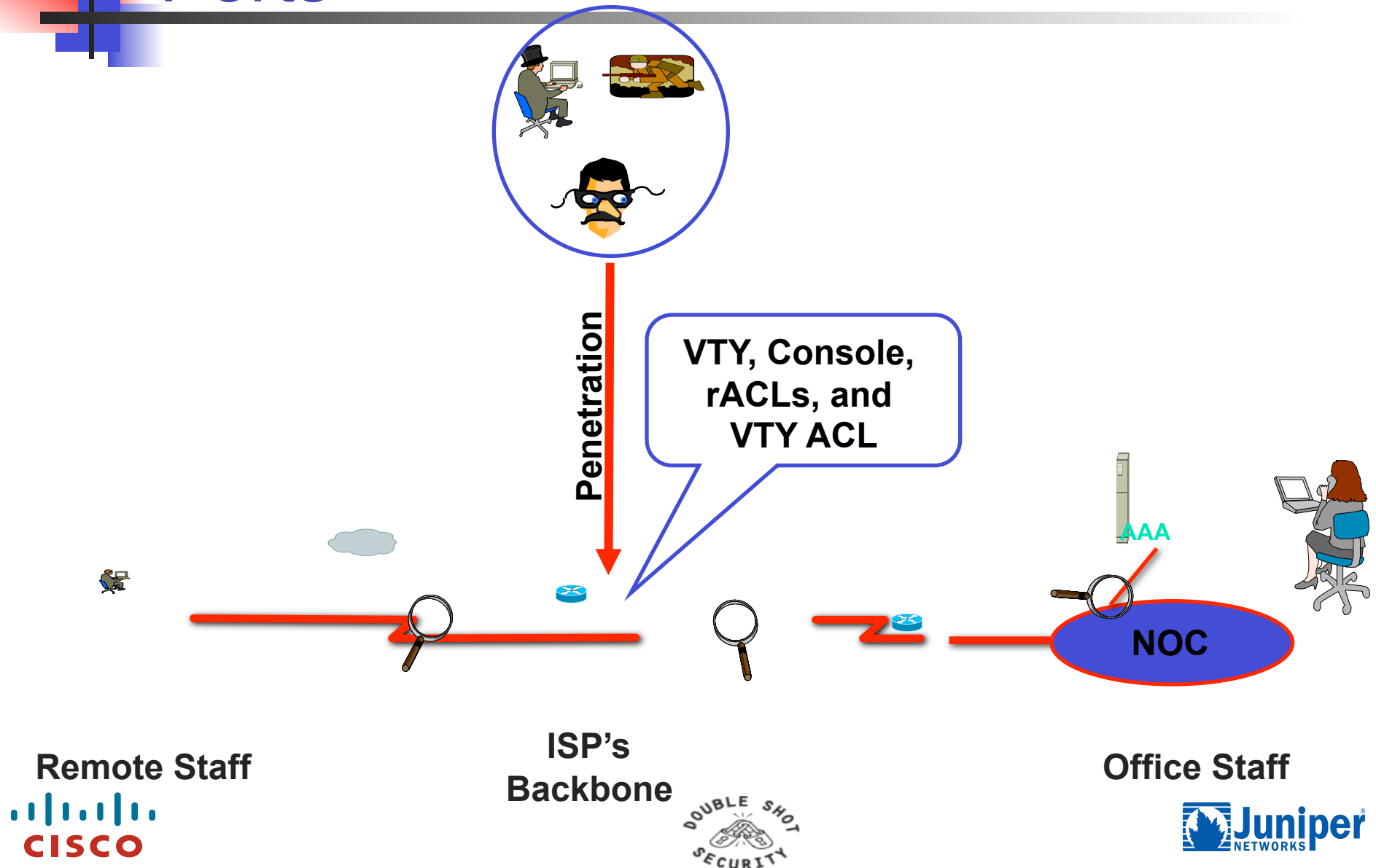


Check List

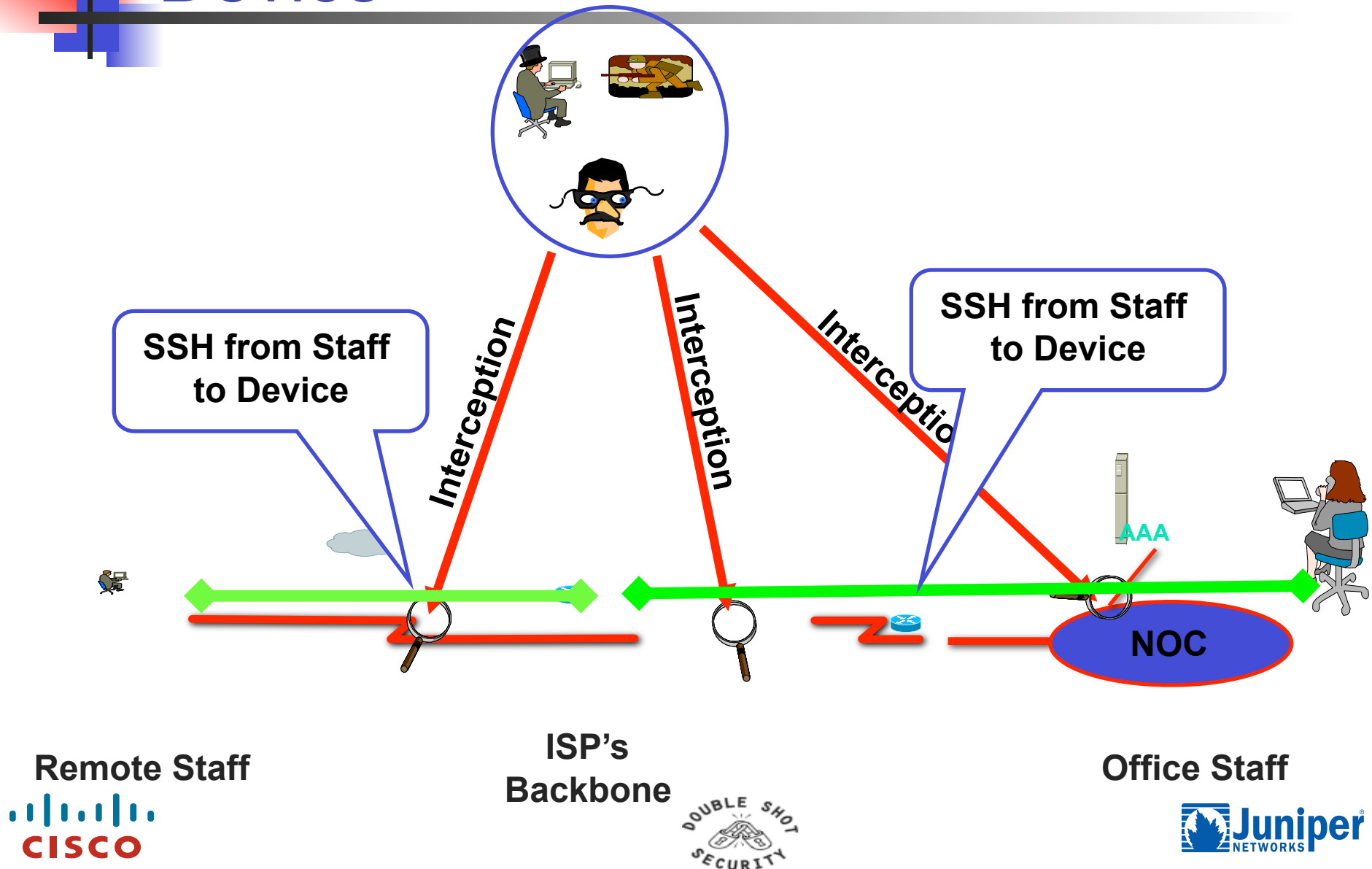
- AAA to the Network Devices
- Controlling Packets Destined to the Network Devices
- Config Audits



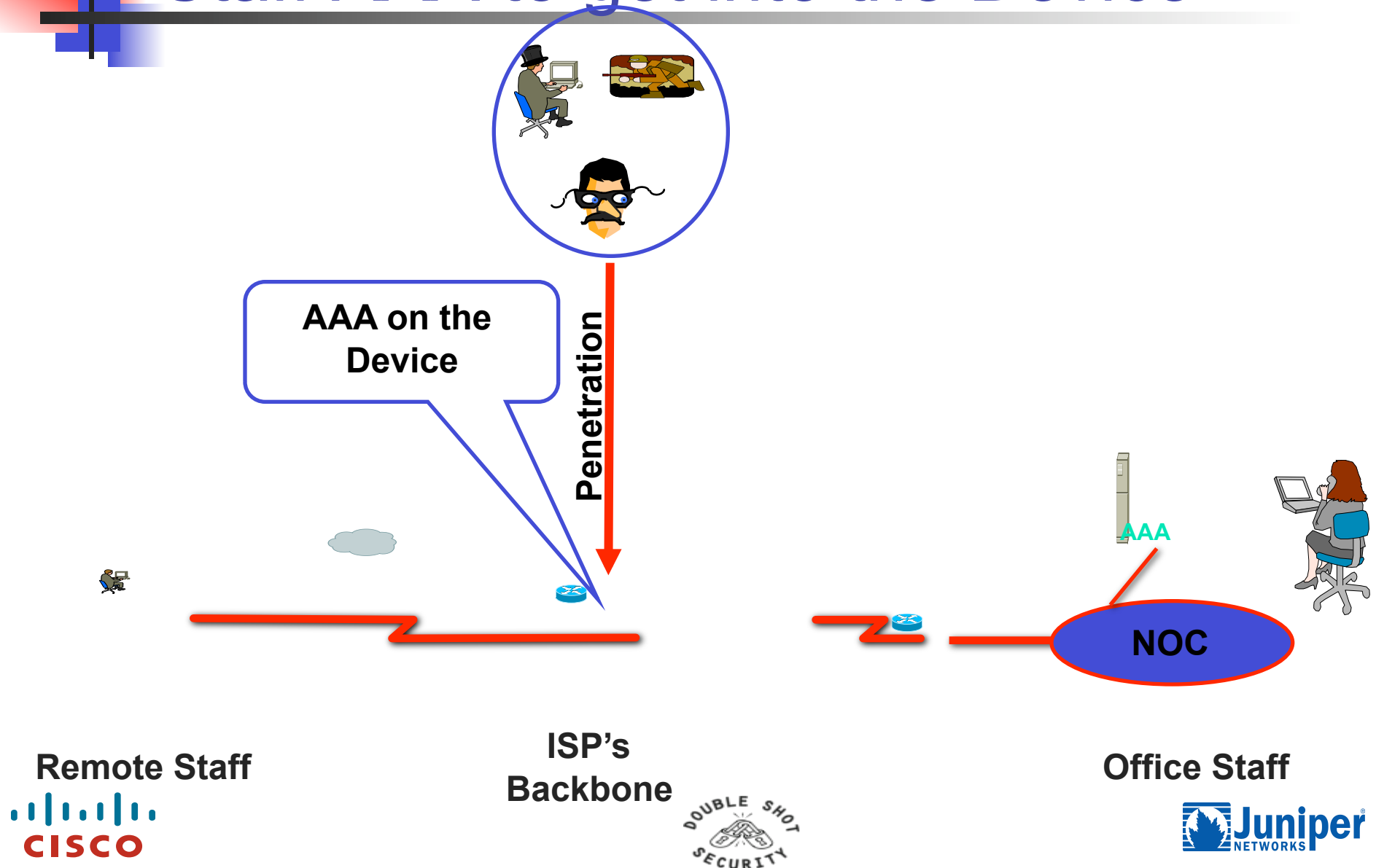
Lock Down the VTY and Console Ports



Encrypt the Traffic from Staff to Device

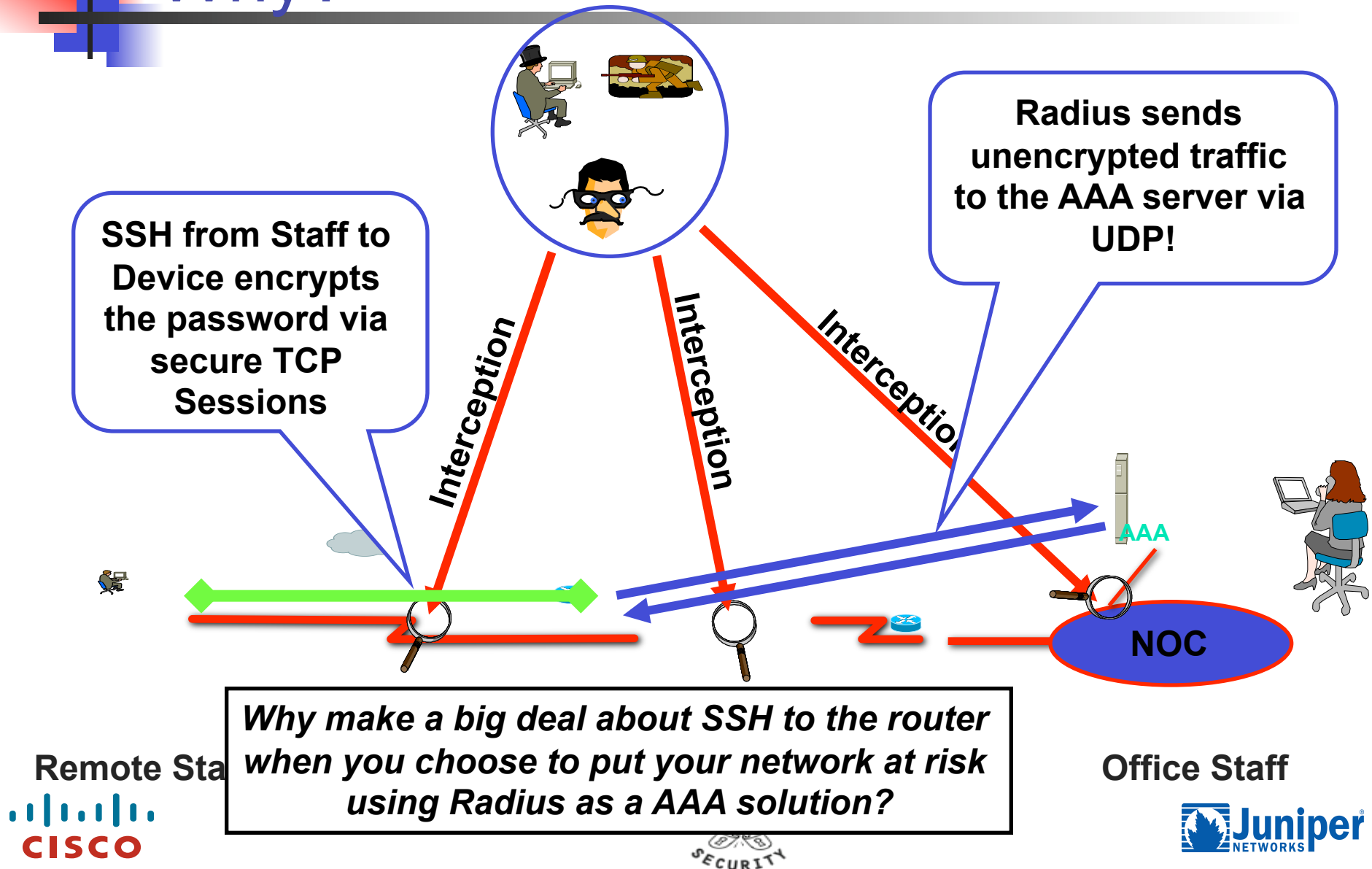


Staff AAA to get into the Device

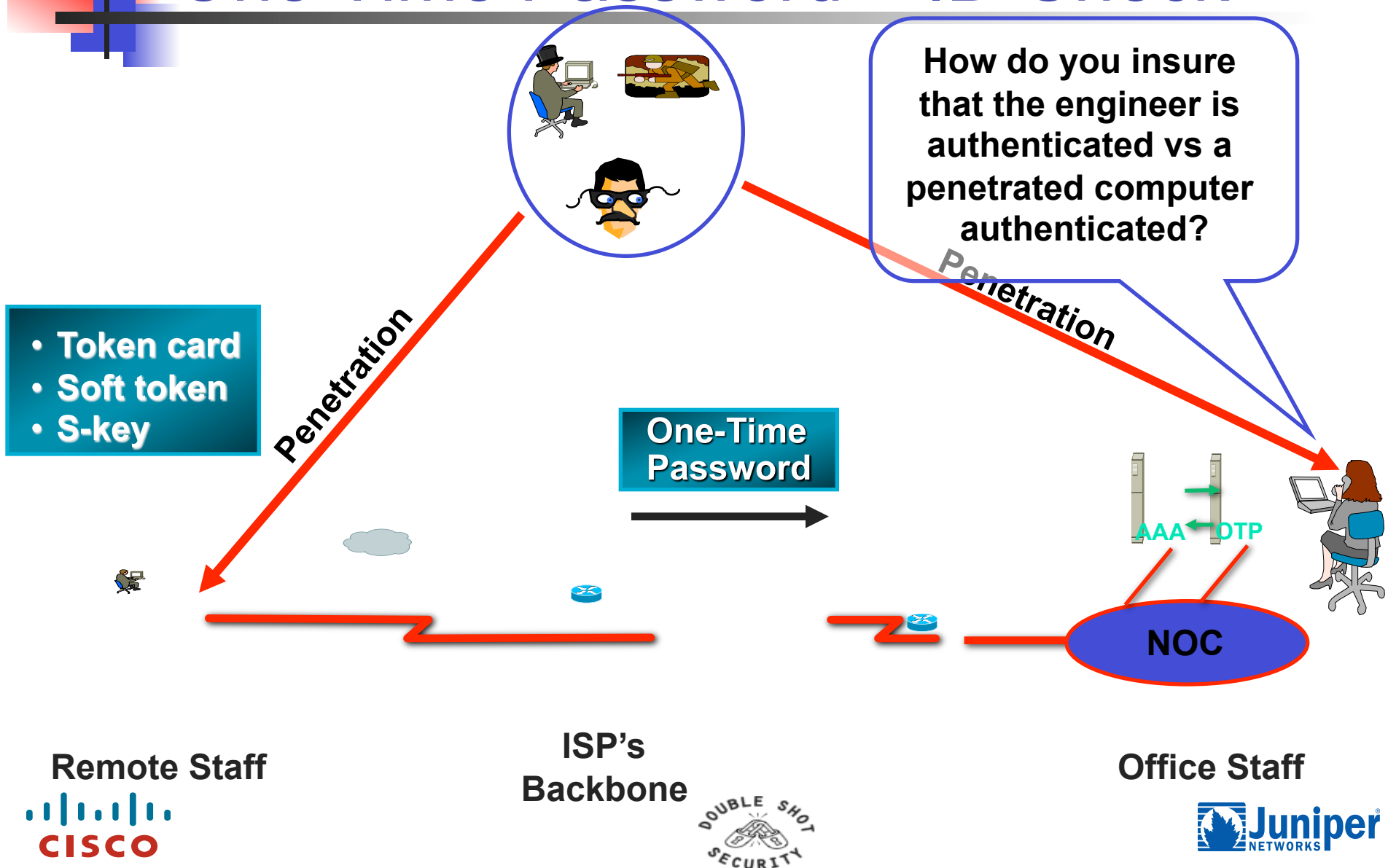


Radius is not an ISP AAA Option!

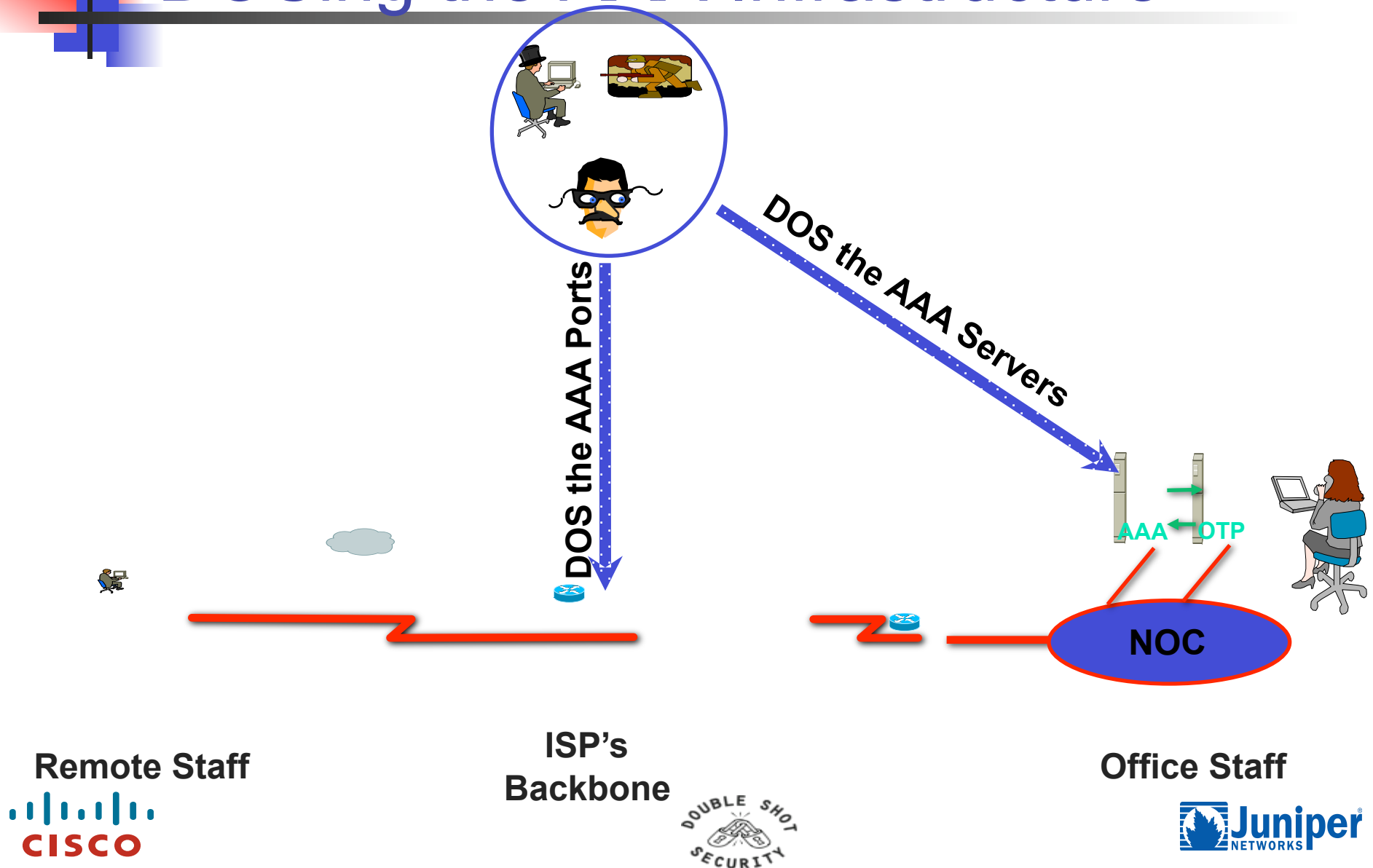
Why?



One Time Password – ID Check

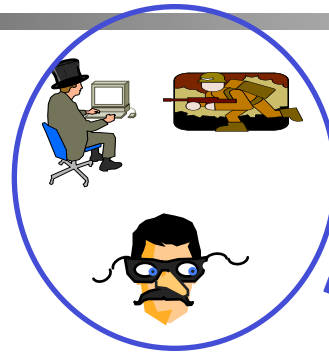


DOSing the AAA Infrastructure



Use a Firewall to Isolate the AAA Servers

Statefull inspection is another reason to select TCP base AAA over UDP.



DOS the AAA Ports

DOS the AAA Servers

Separate AAA Firewall to protect from internal and external threats.

AAA ← OTP



NOC

NOC Firewall

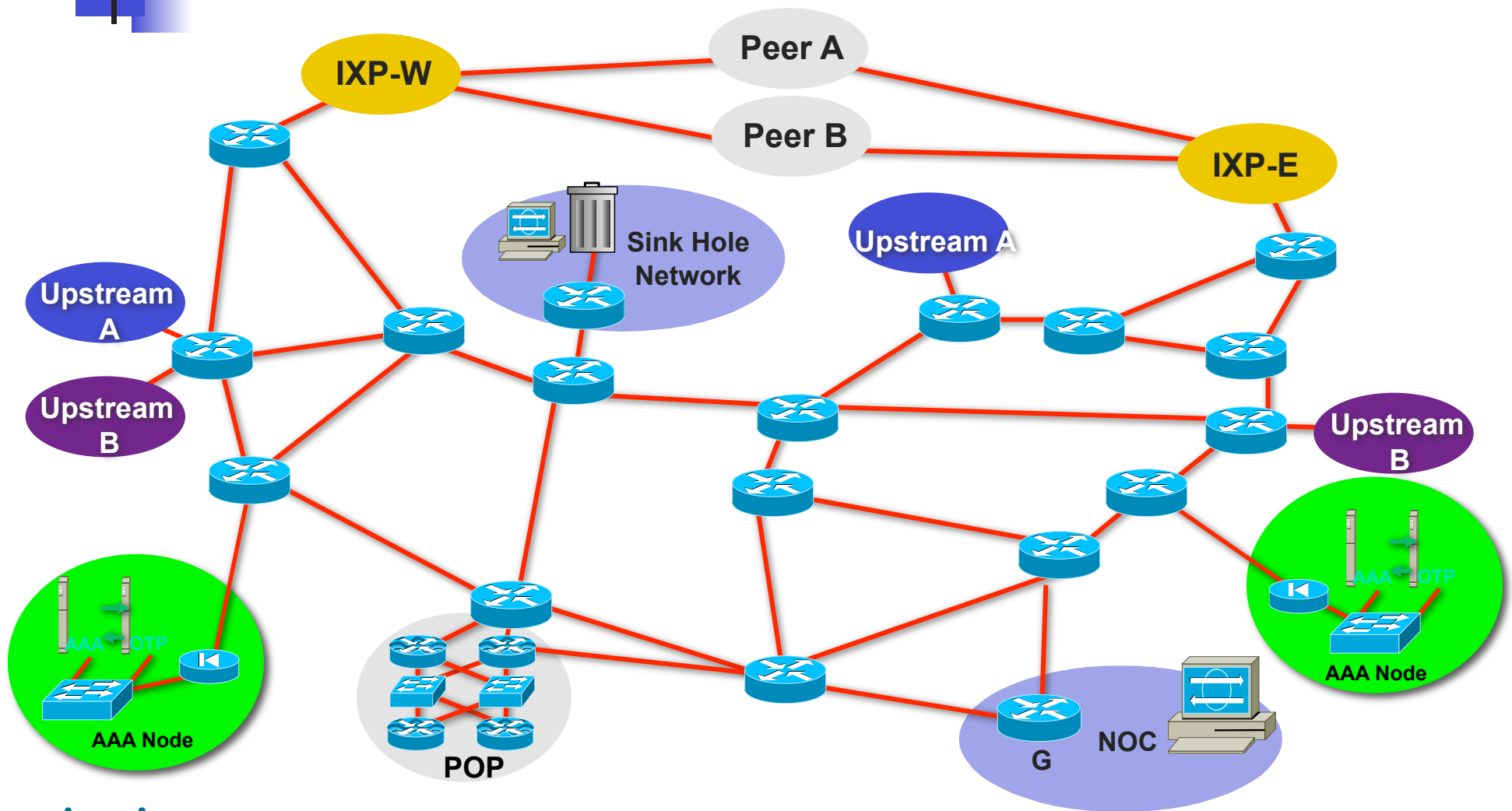
Office Staff

ISP's Backbone

Remote Staff



Distribute AAA Servers and Config Backup

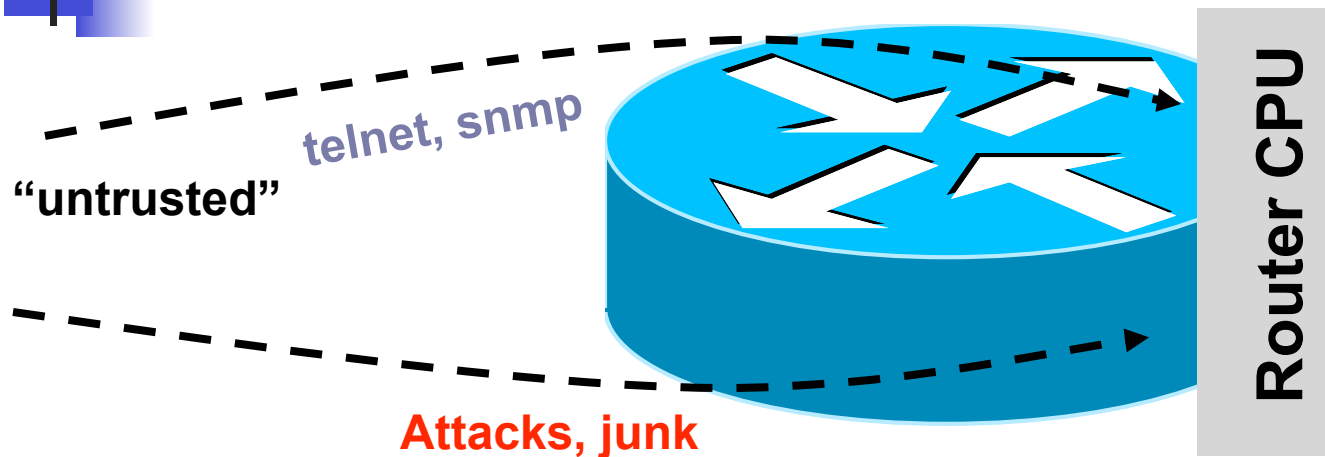




TACACS+ URLs

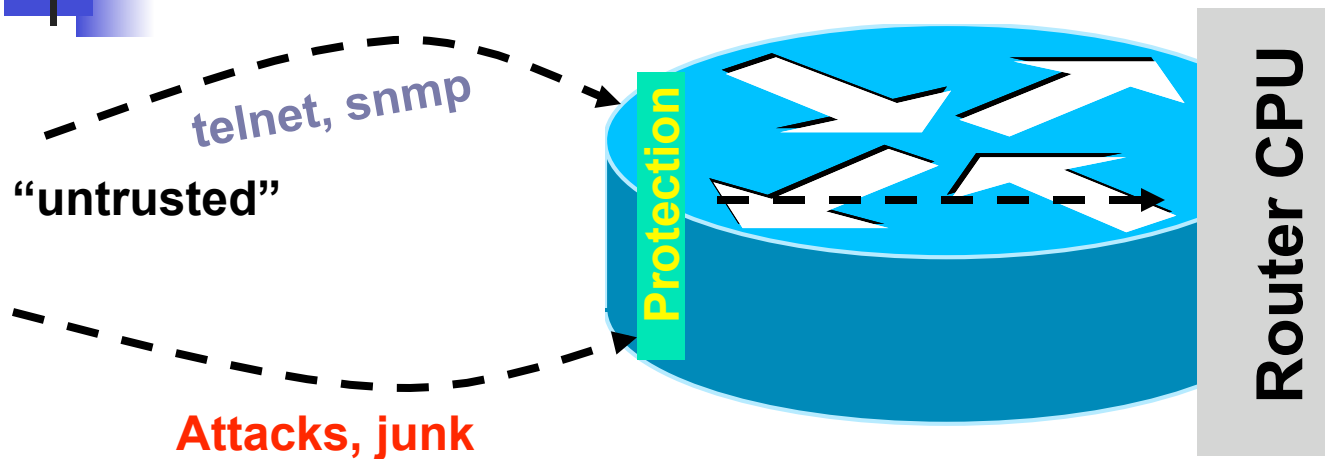
- TACACS+ Open Source
 - <ftp://ftp-eng.cisco.com/pub/tacacs/>
 - Includes the IETF Draft, Source, and Specs.
- Extended TACACS++ server
 - <http://freshmeat.net/projects/tacpp/>
- TACACS + mods
 - http://www.shrubbery.net/tac_plus/

The Old World: Router Perspective

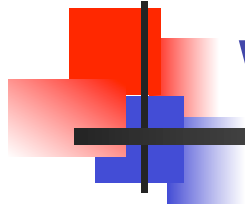


- Policy enforced at process level (VTY ACL, SNMP ACL, etc.)
- Some early features such as ingress ACL used when possible

The New World: Router Perspective



- Central policy enforcement, prior to process level
- Granular protection schemes
- On high-end platforms, hardware implementations



Watch the Config!

- There has been many times where the only way you know someone has violated the router is that a config has changed.
- Of course you need to be monitoring your configs.



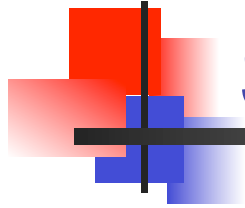
Config Monitoring



- **RANCID - Really Awesome New Cisco config Differ (but works with lots of routers)**
<http://www.shrubbery.net/rancid/>
<http://www.nanog.org/mtg-0310/rancid.html>
- **Rancid monitors a device's configuration (software & hardware) using CVS.**
- **Rancid logs into each of the devices in the device table file, runs various show commands, processes the output, and emails any differences from the previous collection to staff.**



Controlling access



Securing the Box

- Identify what services to what networks
 - Classic prudent policy
 - Allow what you know
 - Then deny all!



Securing Physical Access

- No method of configuring a system can protect it if physical access is not secure
- Attackers with physical access can:
 - Physically harm and degrade the system
 - Perform password recovery and obtain access to the CLI
 - Attach a tap and packet sniffer to obtain traffic captures
 - Do other nefarious things limited only by their imagination...

Protect your systems from unauthorized physical access!



Protecting the Diagnostics Port

```
[[edit system]
```

```
lab@r5# set diag-port-authentication ?
```

Possible completions:

```
+ apply-groups          Groups from which to inherit configuration data
```

```
  encrypted-password    Crypted password string
```

```
  plain-text-password   Prompt for plain text password (auto-crypted)
```

```
[edit system]
```

■ Passwords

- Use `plain-text-password` to enter password directly
 - Stored as an MD5 hash in the configuration
- Use `encrypted-password` to paste in an existing MD5 hash



Securing Logical Access

- Logical access is more difficult to secure
 - Attackers do not require physical access to logically access the system
- Terminal servers can provide backdoor. For you or an attacker
 - Some extra logins/ features are available from console
- One of many ways to protect your systems from unauthorized logical access is to secure it with user names and passwords



Agenda: Configuring Root Authentication

- Why Secure CLI Access Is Needed
 - ➔ Configuring Root Authentication
- Configuring Login Users and Classes
 - Creating Users
 - Creating Login Classes
 - Setting the Idle Timeout



Root Authentication

- By default, Juniper Networks routers have only a single user configured, called *root*
 - Juniper Networks routers do not have a default password configured for the root account.
- Cisco has no console login password or enable password
- Systems with root accounts with no passwords do not last long on the Internet

Configure the root account with a difficult-to-guess password as the first step in securing the system!



Agenda: Configuring Login Users and Classes

- Why Secure CLI Access Is Needed
- Configuring Root Authentication
- Configuring Login Users and Classes
 - Creating Users
 - Creating Login Classes
 - Setting the Idle Timeout



Creating Users

```
[edit]
```

```
lab@R5# edit system login user jsmith
```

```
[edit system login user jsmith]
```

```
lab@R5#
```

Adding a user involves only creating the appropriate container in the configuration



Setting the Full Name

```
[edit system login user jsmith]
```

```
lab@R5# set full-name "John Smith"
```

```
[edit system login user jsmith]
```

```
lab@R5# show
```

```
full-name "John Smith";
```

Optionally, you can configure a text string to identify this user



Setting the Password

```
[edit system login user jsmith]
```

```
lab@R5# set authentication plain-text-password
```

```
New password: extremely tough password
```

```
Retype new password: extremely tough password
```

```
[edit system login user jsmith]
```

```
lab@R5# show
```

```
full-name "John Smith";
```

```
authentication {
```

```
    encrypted-password "$1$wrcIE7//
```

```
    $61gsASq1vP90ktkPgpiCz0"; # SECRET-DATA
```

**Stored in the configuration
as an MD5 hashed value**



Attaching Users to Login Classes

```
[edit system login user jsmith]
```

```
lab@R5# set class ?
```

Possible completions:

<class>	Login class
operator	
read-only	
super-user	
superuser	
unauthorized	

- A user must be a member of one (and only one) login class
 - Preconfigured login classes are available



Preconfigured Login Classes

Class	Permission Bits Set
super-user superuser (Identical functionality)	All
read-only	View
operator	Clear, Network, Reset, Trace, View
unauthorized	None



Available Permissions (1 of 2)

```
[edit system login class restricted-operator]
```

```
lab@R5# set permissions ?
```

Possible completions:

[Open a set of values
admin	Can view user accounts
admin-control	Can modify user accounts
all	All permission bits turned on
clear	Can clear learned network information
configure	Can enter configuration mode
control	Can modify any configuration values
edit	Can edit full files
field	Special for field (debug) support
firewall	Can view firewall configuration
firewall-control	Can modify firewall configuration
floppy	Can read and write the floppy drive
interface	Can view interface configuration
interface-control	Can modify interface configuration



Available Permissions (2 of 2)

maintenance	Can perform system maintenance (as wheel)
network	Can access the network
reset	Can reset and restart interfaces and
rollback	Can rollback for depth greater than zero
routing	Can view routing configuration
routing-control	Can modify routing configuration
secret	Can view secret configuration
secret-control	Can modify secret configuration
security	Can view security configuration
security-control	Can modify security configuration
shell	Can start a local shell
snmp	Can view SNMP configuration
snmp-control	Can modify SNMP configuration
system	Can view system configuration
system-control	Can modify system configuration
trace	Can view trace file settings
trace-control	Can modify trace file settings
view	Can view current values and statistics



Allow and Deny Commands

```
[edit system login class restricted-operator]
lab@R5# set allow-commands "clear bgp"
```

```
[edit system login class restricted-operator]
lab@R5# set deny-commands "telnet"
```

```
[edit system login class restricted-operator]
lab@R5# show
permissions [ network trace view ];
allow-commands "clear bgp";
deny-commands telnet;
```

■ More options:

- Allow commands permit additional access beyond that allowed by permissions
- Deny commands restrict access normally allowed by permissions

Regular Expressions—Commands

```
[edit system login class restricted-operator]
lab@R5# set allow-commands "clear ospf|clear bgp"
```

```
[edit system login class restricted-operator]
lab@R5# show
permissions [ network trace view ];
allow-commands "clear ospf|clear bgp";
deny-commands telnet;
```

Operator	Match ...
	One of the two terms
^	Beginning of the expression
\$	End of the expression
[]	Range of letters or digits
()	Group of commands

Regular Expressions—Example 1

```
[edit]
system {
  login {
    class operator-may-reboot {
      permissions [ clear network reset trace view ];
      allow-commands "request system reboot";
    }
  }
}
```

Class has operator privileges and can reboot the system

Operator	Match ...
	One of the two terms
^	Beginning of the expression
\$	End of the expression
[]	Range of letters or digits
()	Group of commands



Regular Expressions—Example 2

```
[edit]
system {
  login {
    class may-not-set {
      permissions [ clear network reset trace view ];
      deny-commands "^set";
    }
  }
}
```

Class has operator privileges but cannot use any command beginning

Operator	Match ...
	One of the two terms
^	Beginning of the expression
\$	End of the expression
[]	Range of letters or digits
()	Group of commands

Regular Expressions— Configuration

```
[edit system login class restricted-configuration]
lab@R5# set deny-configuration "(system login class) | (system services)"
[edit system login class restricted-configuration]
lab@R5# show
permissions configure;
allow-commands "(show bgp summary | show ospf neighbor)";
deny-configuration "(system login class | system services)";
```

Operator	Match ...
	One of the two terms
^	Beginning of the expression
\$	End of the expression
[]	Range of letters or digits
()	Group of commands



IOS Role-Based CLI Views commands

- Enable AAA using the 'aaa new-model' global config command
 - `aaa new-model`
- Configure the AAA default list to use the router's local database for authentication and authorization
 - `aaa authentication login default local`
 - `aaa authorization exec default local`
- Configure AAA console authorization
 - `aaa authorization console`
- Access the root view. You need to first configure a secret or enable password before you can access the root view. configure a secret password = cisco)
 - `Edge_C38#conf t`
- Enter configuration commands, one per line. End with CNTL/Z.
 - `Edge_C38(config)#enable secret cisco`
 - `Edge_C38(config)#^Z`
 - `Edge_C38#`
 - `Edge_C38#enable view`
 - Password: *secret_password*



IOS Role-Based CLI Views commands

- Edge_C38#
- Create the Operator view
- Configure a password for this view
 - Ping
 - Show controllers
 - Show interfaces
 - Show version
- parser view operator
 - password 5 opspassword
 - commands exec include ping
 - commands exec include show version
 - commands exec include show controllers
 - commands exec include show interfaces



Setting the Idle Timeout

```
[edit system login class restricted-operator]  
lab@R5# set idle-timeout 10
```

JUNOS

```
R5(config)# line vty 0 4
```

```
R5(config)# exec timeout 0 10
```

IOS

- No idle-timeout by default
 - Set the time, in minutes, after which an idle user is automatically disconnected
 - User session is sent warning messages 5 minutes, 1 minute, 10 seconds, and on session shutdown



Is Any Remote Access Secure?

- No remote access to the router is completely secure
 - The issue:
 - Nothing is ever completely secure
 - We need remote access to the router
 - We must minimize the risk
 - The defaults:
 - CLI access is available only on the console port
 - No access is available on the auxiliary port without configuration
 - No other method of remote access is available without configuration



Remote Access Methods

- Remote access methods available on most platforms:
 - Remote access to the CLI
 - Telnet (Client and server)
 - SSH (Client and server)
 - Rlogin (Server, client available in the shell, JUNOS)
 - Remote access to the file system, (JUNOS)
 - FTP (client and server)
 - SCP (client and server)
 - Other remote access
 - Finger (server, client available in the shell)
 - SNMP



Telnet and FTP Servers

- Telnet and FTP
 - Provide convenient access to the CLI and file system
 - Both the Telnet and FTP Servers are disabled by default
 - Everything transmitted (including the password) is sent in cleartext on the wire
 - Custom packet sniffers are available to search for and reassemble the passwords in a both Telnet and FTP sessions
 - The root user can never log in with Telnet or FTP
 - Both protocols provide only *availability*
 - *Confidentiality* and *integrity* are not protected



Enabling the Telnet Server

- The following command enables Telnet access
 - Disabled by default

JUNOS

```
[edit system]
```

```
lab@R1# set services telnet
```

```
[edit system]
```

```
lab@R1# show
```

```
services {  
    telnet;
```

IOS

```
R1 (config) # line vty 0 4
```

```
R1 (config) # login
```

```
R1 (config) # password Uj%  
$3
```



Additional Telnet Options

Options:

- The connection limit establishes the maximum number of concurrent sessions (JUNOS default = 75) (IOS default = 5)
- The rate limit establishes the maximum number of connections allowed per minute (JUNOS default = 150)



Enabling the FTP Server

- The following command enables the FTP server
 - Disabled by default

- `[edit system]`

```
lab@R1# set services ftp
```

```
[edit system]
```

```
lab@R1# show
```

```
services {
```

```
    ftp;
```

```
}
```



SSH Clients

- Many SSH implementations are available
 - Putty
 - TeraTerm
 - OpenSSH
 - Many commercial servers and clients



Enabling the SSH Server

- The following command enables SSH access
 - Disabled by default

```
[edit system]
```

```
lab@R1# set services ssh
```

```
[edit system]
```

```
lab@R1# show
```

```
services {  
    ssh;
```



Allowing Root Logins

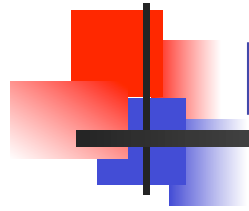
[edit]

lab@R1# set system services ssh root-login ?

Possible completions:

allow	Allow root access via ssh
deny	Do not allow root access via ssh
deny-password only	Allow for non-password-based authentication methods

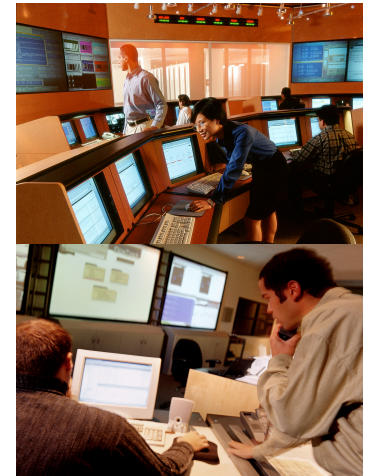
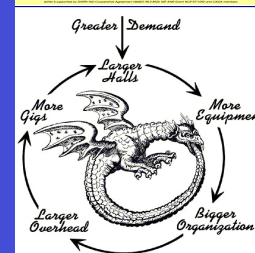
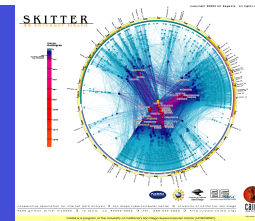
- By default, once SSH access is turned on, user *root* cannot log in with SSH
 - User *root* can be allowed to log in:
 - Normally, with password-based authentication
 - Only with key-based authentication
 - Is this something you really want to do?



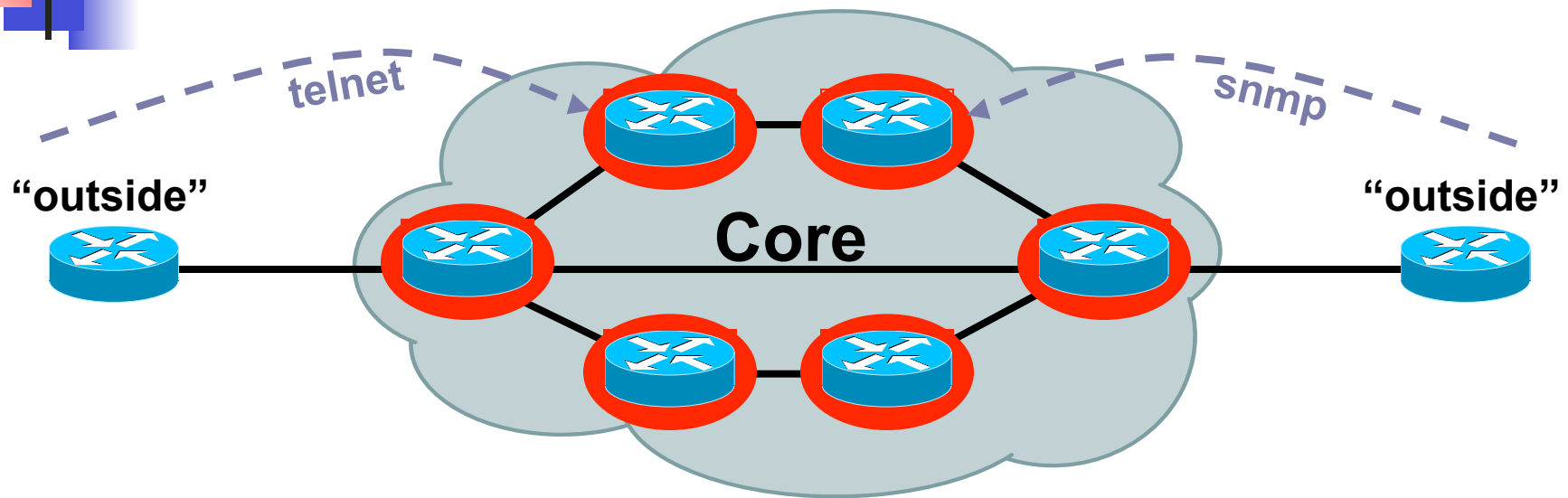
Lab – Securing remote access

- Follow the lab guide

Edge Protection

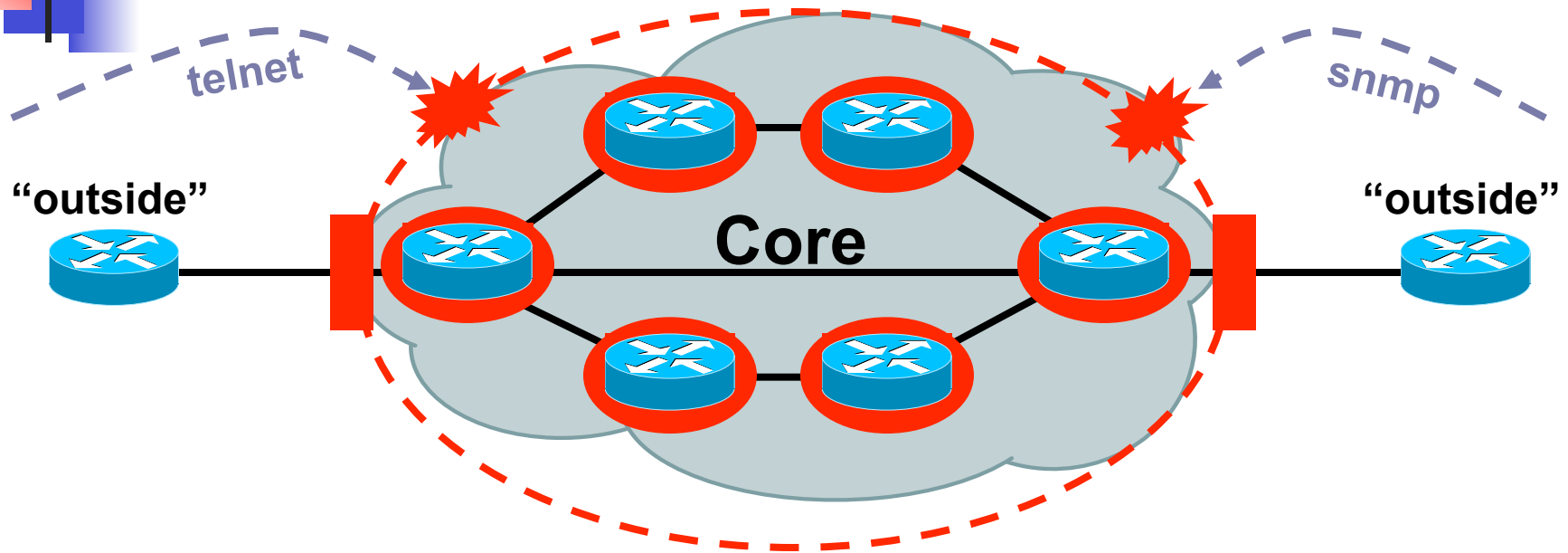


The Old World: Network Edge



- Core routers individually secured
- Every router accessible from outside

The New World: Network Edge



- Core routers individually secured PLUS
- Infrastructure protection
- Routers generally NOT accessible from outside



Infrastructure ACLs

- Basic premise: filter traffic destined TO your core routers
 - Do your core routers really need to process all kinds of garbage?
- Develop list of required protocols that are sourced from outside your AS and access core routers
 - Example: eBGP peering, GRE, IPSec, etc.
 - Use classification ACL as required
- Identify core address block(s)
 - This is the protected address space
 - Summarization is critical → simpler and shorter ACLs



Infrastructure ACLs

- Infrastructure ACL will permit only required protocols and deny ALL others to infrastructure space
- ACL should also provide anti-spoof filtering
 - Deny your space from external sources
 - Deny RFC1918 space
 - Deny multicast sources addresses (224/4)
 - RFC3330 defines special use IPv4 addressing



A Digression: IP Fragments and Security

- Fragmented Packets can cause problems...
 - Fragmented packets can be used as an attack vector to bypass ACLs
 - Fragments can increase the effectiveness of some attacks by making the recipient consume more resources (CPU and memory) due to fragmentation reassembly

A Digression: IP Fragments and Security

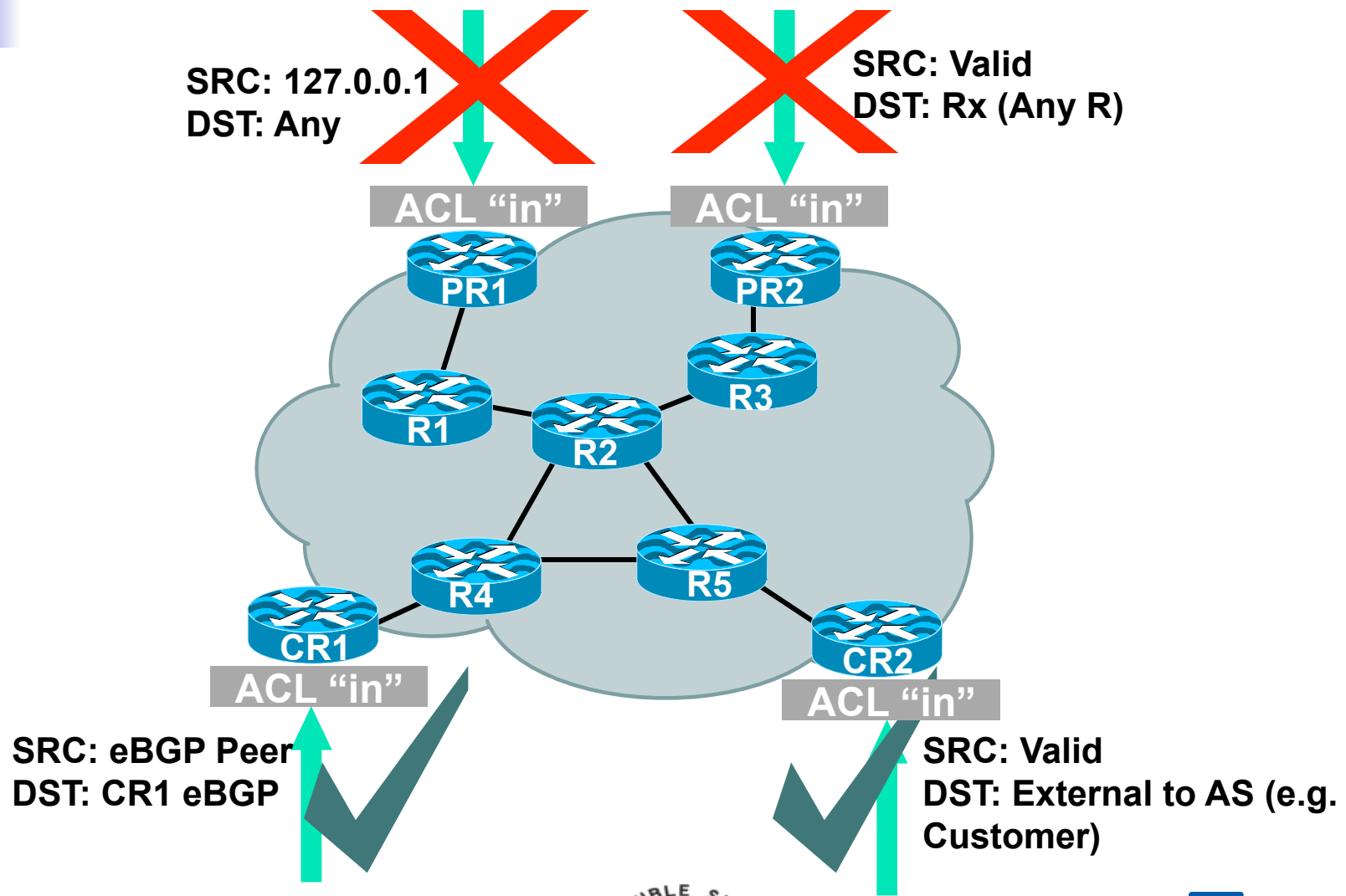
- By default (without the **fragments** keyword)...
 - Initial fragments and non-fragmented packets
 - L3 ACLs—access control entry (ACE) action executed (permit/deny) available L3 information
 - L4 ACLs—ACE action executed (permit/deny) available L4 information
 - Non-initial fragment packets (assuming L3 match)
 - L3 ACLs—ACE action executed (permit/deny) available L3 information
 - L4 ACLs—ACE action executed (permit/deny) based on L3 info (there is no L4 info in the fragment) and protocol **only**
- The ACL **fragments** keyword enables specialized handling behavior
 - Initial fragments and non-fragmented packets
 - L3 and L4 ACLs—the packet does not match the entry since the fragment keyword is used. The packet then “falls through” to the next line(s)
 - Non-initial fragment packets (assuming L3 match)
 - With L3 and L4 ACLs—with an L3 match (and protocol matches the IP protocol), the action of the ACE is executed (permit/deny)



Infrastructure ACLs

- Infrastructure ACL must permit transit traffic
 - Traffic passing through routers must be allowed via permit IP any any
- ACL is applied inbound on ingress interfaces
- Fragments destined to the core can be filtered via fragments keyword

Infrastructure ACL in Action





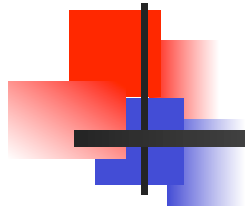
IP Options

- Provide control functions that may be required in some situations but unnecessary for most common IP communications
- IP Options not switched in hardware
- Complete list and description of IP Options in RFC 791
- Drop and ignore reduce load on the route processor (RP)
- Caution: some protocols/application require options to function:
 - For example: strict/loose source routing, resource reservation protocols (RSVP) and others
- `ip access-list extended drop-ip-option`
 - `deny ip any any option any-options`
 - `permit ip any any`



IP Options

- ip options drop
- ip options ignore—router ignores options
 - Best practice when router doesn't need to process options
 - "ignore" not available on all routing platforms
 - Available in 12.0(22)S, 12.3(4)T and 12.2(25)S
http://www.cisco.com/en/US/products/sw/iosswrel/ps1829/products_feature_guide09186a00801d4a94.html



Iterative Deployment

- Typically a very limited subset of protocols needs access to infrastructure equipment
- Even fewer are sourced from outside your AS
- Identify required protocols via classification ACL
- Deploy and test your ACLs



Step 1: Classification

- Traffic destined to the core must be classified
- NetFlow can be used to classify traffic
 - Need to export and review
- Classification ACL can be used to identify required protocols
 - Series of permit statements that provide insight into required protocols
 - Initially, many protocols can be permitted, only required ones permitted in next step
 - Log keyword can be used for additional detail; hits to ACL entry with **log will increase CPU utilization**: impact varies by platform
- Regardless of method, unexpected results should be carefully analyzed → **do not permit protocols that you can't explain!**



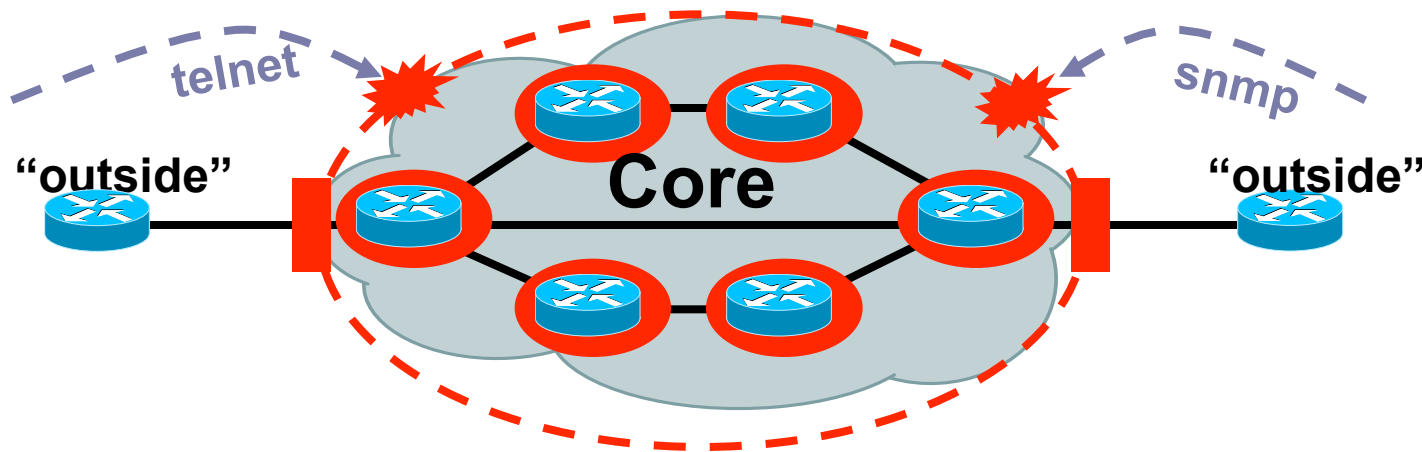
Step 2: Begin to Filter

- Permit protocols identified in step 1 to infrastructure only address blocks
- Deny all other to addresses blocks
 - Watch access control entry (ACE) counters
 - Log keyword can help identify protocols that have been denied but are needed
- Last line: permit ip any any ← permit transit traffic
- The ACL now provides basic protection and can be used to ensure that the correct suite of protocols has been permitted

Steps 3 and 4: Restrict Source Addresses

- Step 3:
 - ACL is providing basic protection
 - Required protocols permitted, all other denied
 - Identify source addresses and permit only those sources for requires protocols
 - e.g., external BGP peers, tunnel end points
- Step 4:
 - Increase security: deploy destination address filters if possible

Infrastructure ACLs



- Edge "shield" in place
- Not perfect, but a very effective first round of defense
 - Can you apply iACLs everywhere?
 - What about packets that you cannot filter with iACLs?
 - Hardware limitations
- Next step: secure the control/management planes per box



Packet Filters



Packet filter applications

- Protect control plane (RE)
- Limit services
- Drop packets that don't belong
 - Spoofed packets
 - uRPF perhaps better application

Overview of Firewall Filter Syntax

```
[edit firewall family inet]
filter filter-name {
  term term-name {
    from {
      match-conditions;
    }
    then {
      action;
      action-modifiers;
    }
    term implicit-rule {
      then discard;
    }
  }
}
```

- Syntax similar to policy statements
- Defined under [edit firewall family] hierarchy level
- Named filters, one or more terms
 - Terms processed sequentially
 - All packets match a term when a *from* condition is not specified
 - Implicit *discard all* for packets that do not match any term
- Actions: accept, reject, and discard
 - Modifiers: log, count, sample, etc.
- One filter per logical unit, per direction; the same filter can be used on many interfaces



Overview of Match Conditions

- Firewall match conditions
 - Generally, each term in a filter has a match condition
 - Terms without a `from` statement match *all* traffic
 - The `from` statement specifies the conditions the packet must match for the action to be taken
 - Multiple match conditions possible per term
 - All conditions in the `from` statement must match (logical AND)
- Several categories of match conditions
 - Numeric-range filter
 - Address filter
 - Bit-field filter



Numeric Range Filter Match Condition

- Match packet fields that can be identified by a numeric value
 - Port and protocol numbers
- Specify a keyword that identifies the condition and a value that a field in a packet must match
 - `source-port 1024-65535`
 - `source-port smtp`
- Keywords identifying available fields:
 - `destination-port, dscp, fragment-offset, icmp-code, icmp-type, interface-group, packet-length, port, precedence, protocol, source-port`



Address Filter Match Condition

- IP source and destination prefixes
- Keywords available
 - `address prefix` (source or destination)
 - `destination-address prefix`
 - `source-address prefix`
- Address matches are longest *or*



Bit-Field Match Condition

- Match on specific bits in certain packet fields
- You can specify bit fields with symbolic names or numeric values
- Bit matching for IP options, fragment flags, and TCP flags
 - Note: Specification of a bit field does NOT imply the corresponding protocol
- Grouping (...), negation (!), and support for logical AND (& or +), logical OR (| or ,) functions

Bit-Field Match Examples

IP Options

loose-source-route (131)	Strict-source-route (137)
record-route (7)	router-alert (148)
Timestamp (68)	

TCP Flags

ack (0x10)	fin (0x01)	push (0x08)	rst (0x04)
syn (0x02)	urgent (0x20)		

Example: “tcp-flags (0x01 & 0x02)” is equal to “tcp-flags 0x03”

Text Synonyms

first-fragment (matches offset = 0, MF = 1)
tcp-established: Equivalent to “(ack | rst)”
tcp-initial: Equivalent to “(syn & !ack)”



Firewall Actions Overview

- Overview

- Actions fall into two categories
 - Actions: accept, discard, and reject
 - Action modifiers: count, sample, and log/syslog
- Default action is *discard*
 - Use of an action modifier creates an implicit accept (a sampled packet automatically is accepted unless an explicit reject is included in the term)



Action Statements

- Three action statements:
 - `accept`: The packet is accepted for forwarding—no other term is analyzed
 - `reject message-type`: The packet is rejected, and the corresponding ICMP message is generated; no other term is analyzed
 - `discard`: The packet is silently discarded, and no other term is analyzed
 - Provides better security—very useful for DoS attacks due to address spoofing or the use of *zombies*



Reject Message Options

- Based on configuration, the `reject` action generates one of the following:
 - `administratively-prohibited` (default)
 - `bad-host-tos`, `bad-network-tos`
 - `host-prohibited`, `host-unknown`, `host-unreachable`
 - `network-prohibited`, `network-unknown`, `network-unreachable`
 - `port-unreachable`
 - `precedence-cutoff`, `precedence-violation`
 - `protocol-unreachable`
 - `source-host-isolated`, `source-route-failed`
 - `tcp-reset`
 - Generates a TCP reset segment when the rejected traffic is TCP, or else no response is generated



Action Modifiers

- counter-name
 - The corresponding counter is incremented
 - View with `show firewall filter-name`
 - Clear with `clear firewall [all | counter-name | filter-name]`
- Logging
 - The packet sample is logged to the routing daemon cache
 - View with `show firewall log`
 - No clear command—displays most recent entries first
 - Also can log to syslog with `syslog` action
- Sampling
 - Packets are sampled and written to a file based on sampling settings
 - Specified under the `forwarding-options` hierarchy
 - Local ASCII files and `cflowd` version 5|8 export

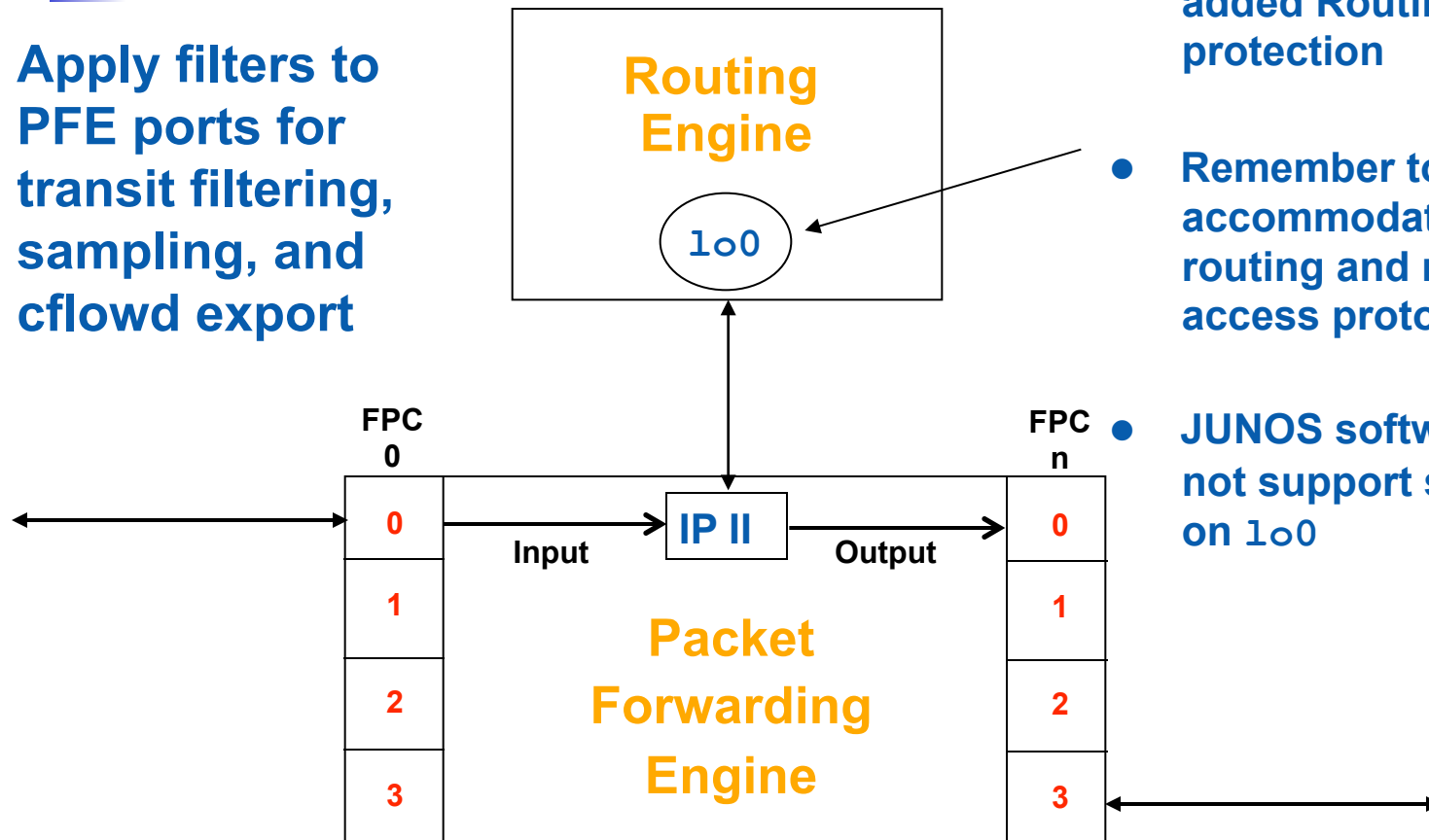
Applying Firewall Filters

```
interfaces {  
    interface-name {  
        unit logical-unit-number {  
            family inet {  
                filter {  
                    input filter-name;  
                    output filter-name;  
                }  
            }  
        }  
    }  
}
```

- Filters must be applied to an interface to take effect
- A common filter can be applied to multiple (or even all) interfaces
- Each interface can support two filters per logical unit—one input and one output
- Apply the filter to the loopback interface for Routing Engine protection

Transit versus Routing Engine Filters

Apply filters to PFE ports for transit filtering, sampling, and cflowd export

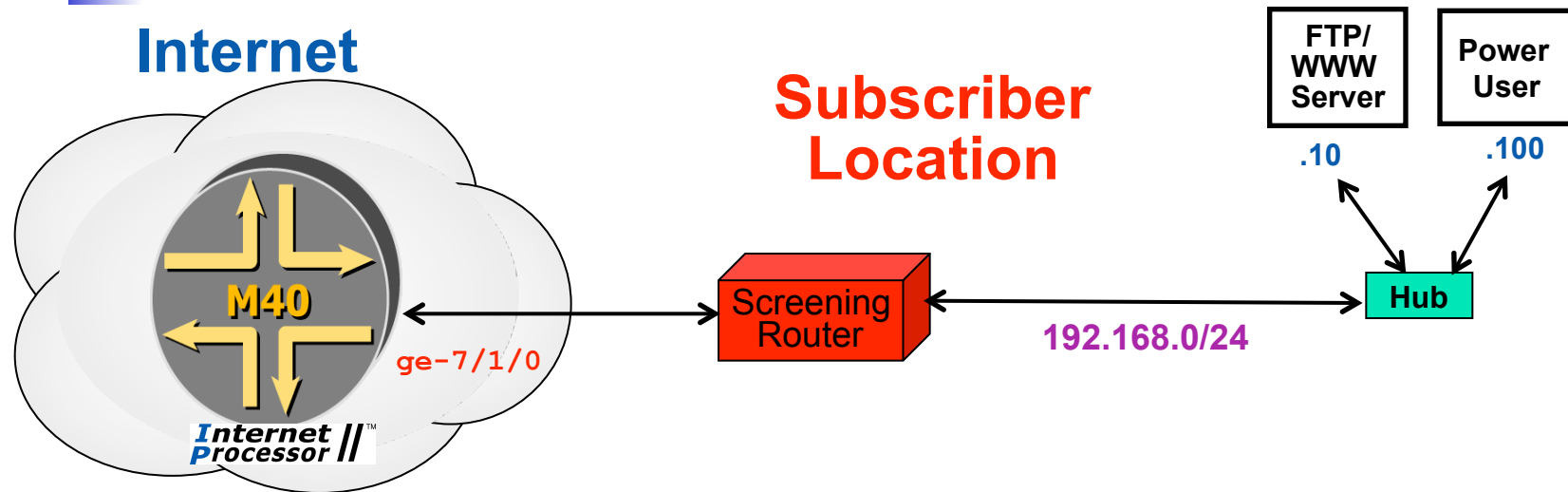


- Apply filters to 100 for added Routing Engine protection

- Remember to accommodate your routing and remote access protocols!

- JUNOS software does not support sampling on 100

Spoof Prevention



Rule 1: Input
From SA = 192.168.0
Then Accept
From SA = other
Then Reject

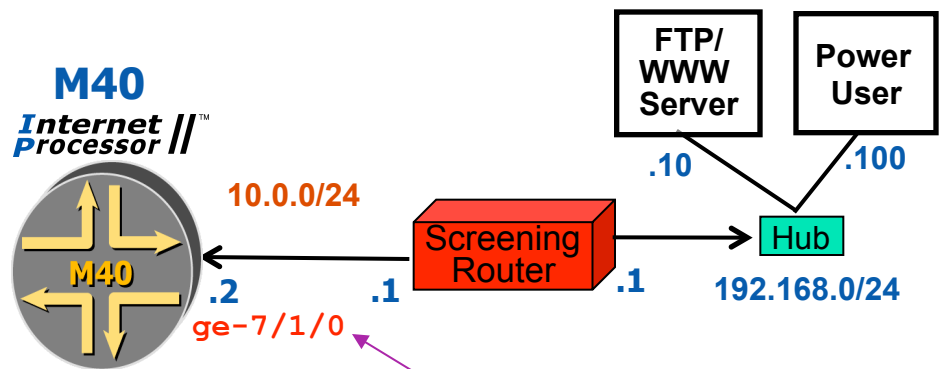
Rule 1 prevents the origination of spoofed packets from this site

Rule 2: Output
From SA = 192.168.0
Then Reject
From SA = other
Then Accept

Rule 2 blocks spoofed packets from entering this site

Inbound Spoof Prevention

```
[edit firewall family inet]
lab@router# show
filter no-spoofs-in {
  term allow-valid {
    from {
      source-address {
        192.168.0.0/24;
        10.0.0.0/24;
      }
    }
    then accept;
  }
  term reject {
    then {
      count bad-source-address;
      log;
      discard;
    }
  }
}
```

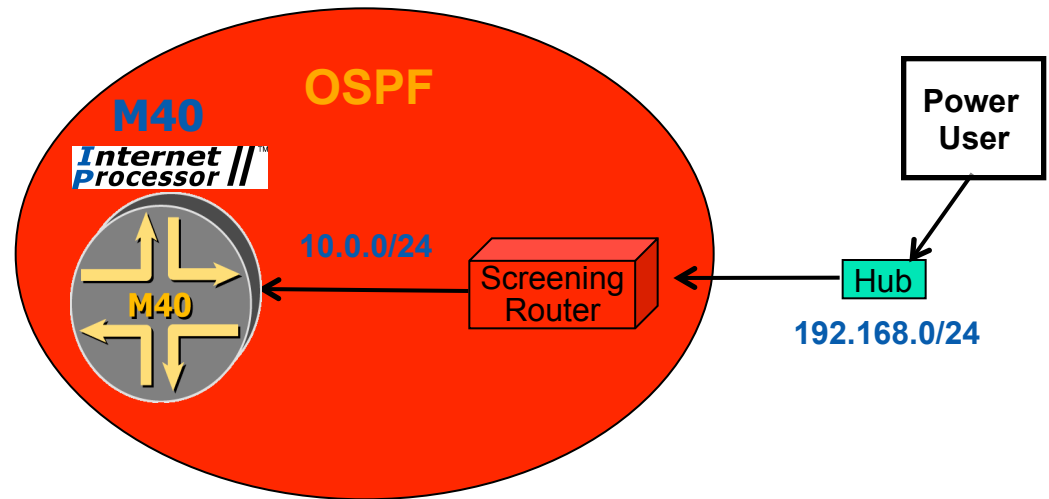


Applied as input filter on subscriber interface

```
[edit interfaces ge-7/1/0]
lab@router# show
unit 0 {
  family inet {
    filter {
      input no-spoofs-in;
    }
    address 10.0.0.2/24;
  }
}
```


Pop Quiz!

```
[edit firewall]
lab@router# show
family inet {
  filter pop-me {
    term telnet {
      from {
        protocol tcp;
        port telnet;
      }
      then accept;
    }
    term ping {
      from {
        protocol icmp;
      }
      then accept;
    }
  }
}
```



Shortly after applying this filter to the 1o0 interface, the user's Telnet session *hangs* and cannot be reestablished

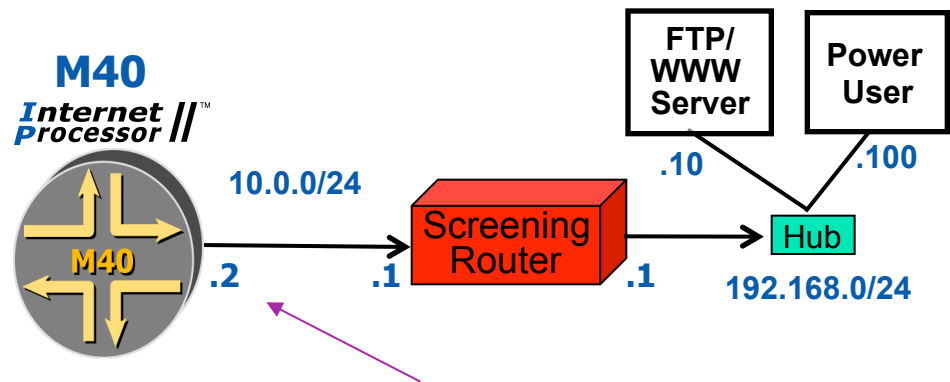
Any ideas?

Preventing Fragmentation Exploits

```
[edit firewall]
lab@San_Jose-3# show
family inet {
  filter no-frags {
    term 1 {
      from {
        is-fragment;
        protocol [ icmp udp ];
      }
      then {
        count no-frags;
        log;
        discard;
      }
    }
    term 2 {
      then accept;
    }
  }
}
```

CISCO

Permits diagnostic pings while blocking fragmented ICMP/UDP traffic
(For example, Teardrop, Boink, POD)



Filter applied in output direction of subscriber interface

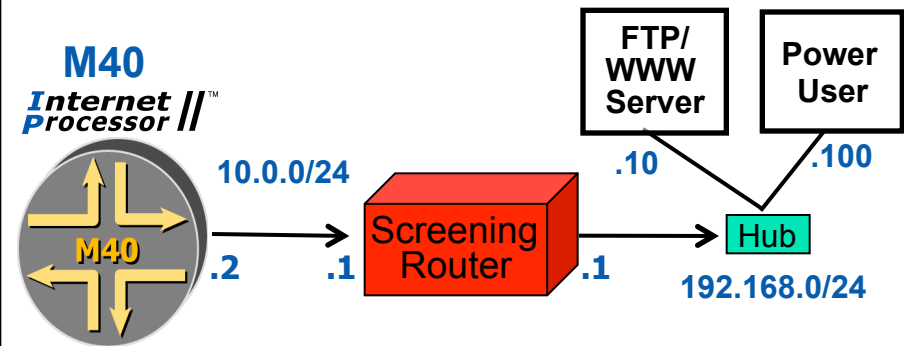
```
[edit interfaces ge-7/1/0]
lab@router# show
unit 0 {
  family inet {
    filter {
      output no-frags;
    }
    address 10.0.0.2/24;
  }
}
```

DOUBLE
SECURITY

iper
NETWORKS

Securing the FTP/WWW Server

```
[edit firewall family inet filter ftp-www-only]
lab@San_Jose-3# show
term allow-ftp-www {
  from {
    destination-address {
      192.168.0.10/32;
    }
    protocol tcp;
    destination-port [ ftp ftp-data http ];
  }
  then accept;
}
term reject-other {
  from {
    destination-address {
      192.168.0.10/32;
    }
  }
  then {
    count unauthorized-service-requests;
    log;
    discard;
  }
}
term accept-all {
  then accept;
}
```



```
interfaces ge-7-1/0 {
  unit 0 {
    family inet {
      filter {
        output ftp-www-only;
      }
    }
  }
}
```

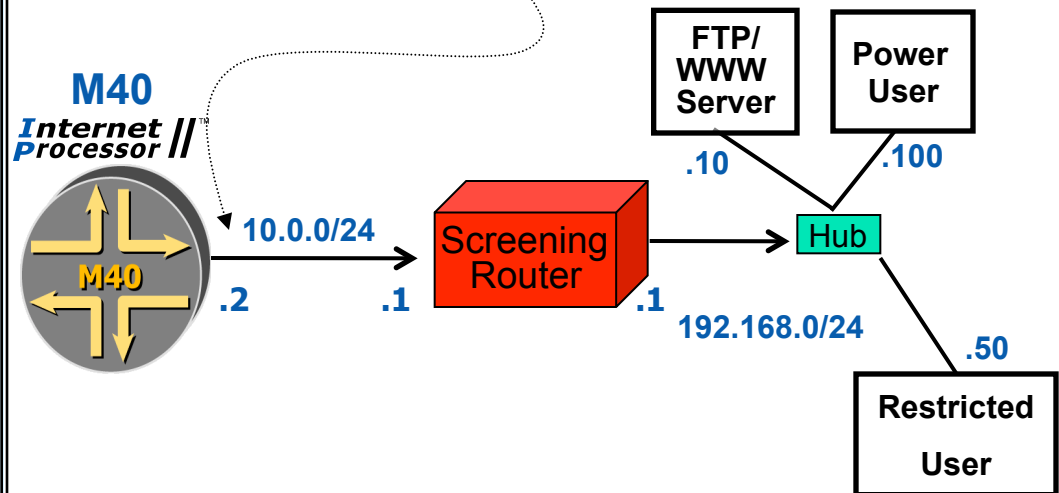
**Filter applied
in output direction of
subscriber interface**

**Remember the implicit *deny all* for unmatched
traffic!**

Outgoing Service Restriction

```
[edit firewall family inet]
lab@San_Jose-3# show filter user-control
term normal-user-allow {
  from {
    destination-address {
      0.0.0.0/0;
      192.168.0.100/32 except;
    }
    protocol tcp;
    source-port http;
    tcp-established;
  }
  then accept;
}
term track-unauthorized {
  from {
    destination-address {
      0.0.0.0/0;
      192.168.0.100/32 except;
    }
  }
  then {
    count unauthorized;
    discard;
  }
}
term power-user {
  then accept;
}
```

Filter applied in output direction to filter response traffic!



Note the use of **except**, which exempts the power user from a particular terms in this filter





Rate Policing

- Instead of allowing or dropping packets that meet match conditions, you can use a filter to identify traffic that is to be policed (rate-limited)
 - You can apply a policer directly to an interface to rate-limit all traffic associated with that protocol family
- Traffic that matches the filter is then policed according to an average bandwidth and a burst size
 - Can specify bandwidth as a percentage of interface speed
- When traffic exceeds the policing parameters, it can:
 - Be discarded
 - Have its loss-priority (PLP) bit set
 - Be associated with a forwarding class (output queue)

Rate Policing Example

```
[edit firewall]
lab@router# show
policer p1 {
    if-exceeding {
        bandwidth-limit 400k;
        burst-size-limit 100k;
    }
    then discard;
}
family inet {
    filter limit-ftp {
        term ftp {
            from {
                source-address {
                    1.2.3.0/24;
                }
                protocol tcp;
                destination-port [ ftp ftp-data ];
            }
            then {
                policer p1;
                count count-ftp;
            }
        }
    }
}
```

● Example:

- **bandwidth-limit**
 - In bits per second
 - 30,520 bps to 4.29 Gbps
- **burst-size-limit**
 - In bytes per second
 - Min should = 10 times MTU (low speed) or bandwidth times 3–5 milliseconds (high speed)
 - Max = 16.7 Mb



Interface-Based Policers

```
interfaces {  
  interface-name {  
    unit logical-unit-number {  
      family inet {  
        filter {  
          input filter-name;  
          output filter-name;  
        }  
        policer {  
          input policer-template;  
          output policer-template;  
        }  
      }  
    }  
  }  
}
```



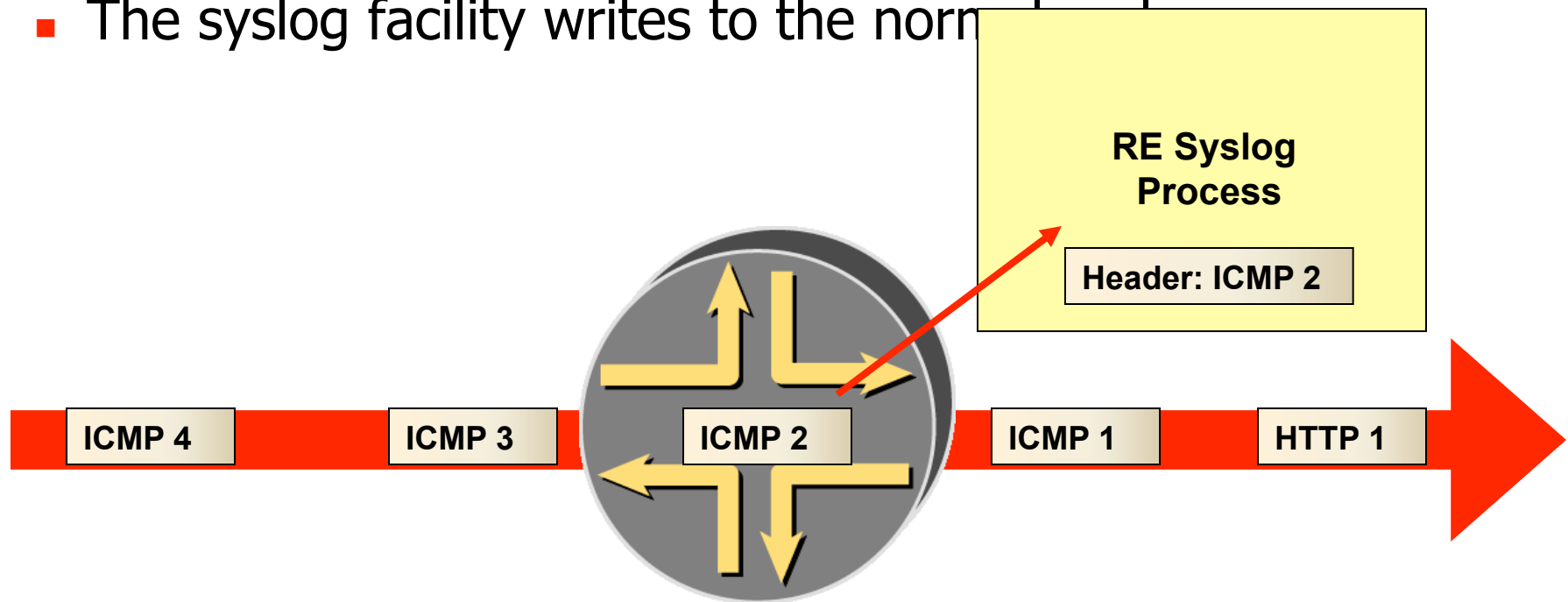
Firewall-Related Operational Commands

- List of commands:

- `show firewall name`
 - Displays counter values
- `show firewall log`
 - Displays kernel log cache
- `show log log-file-name`
 - Displays logged entries when the syslog action modifier is used in a term
- `clear firewall name`
 - Resets counters associated with a firewall
- `show policer`
 - Displays a list of interface policers
- `show interfaces policer interface-name`
 - Displays details about interface policers

Using the Firewall `syslog` Modifier

- Sending alerts to the syslog
 - The `syslog` modifier captures minimal IP information, but allows automated detection and an audit trail
 - The syslog facility writes to the normal



Sample Filter Using Syslog

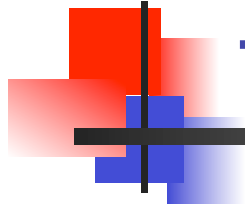
- Logging filter example:
 - Create term to syslog TCP packets with the SYN bit set with a `syslog` action modifier
 - Remember to create a final term!

```
[edit firewall family inet]
lab@R1# show
filter filter-test {
    term count-syn {
        from {
            protocol tcp;
            tcp-initial;
        }
        then {
            syslog;
            accept;
        }
    }
    term others {
        then {
            count other-packets;
            accept;
        }
    }
}
```



Lab: Secure the control plane

- Follow the lab guide

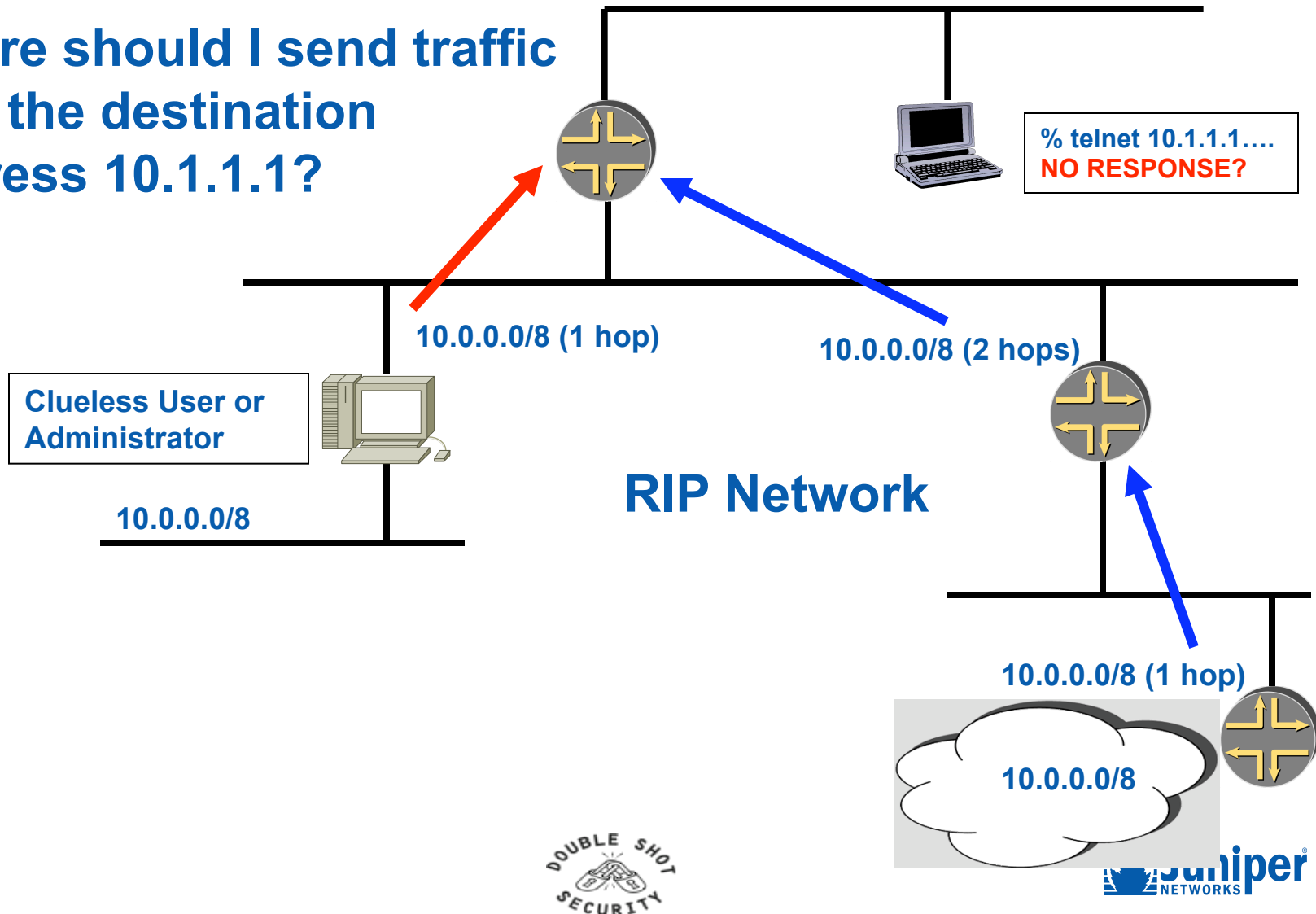


The Need for Secure Routing

- Q: What is the *most important* function performed by routers?
 - A: They route!
- Q: What happens when you lose control of this function?
 - A: A compromised router or routing protocol can result in problems that run the gambit from simple loss of connectivity to a compromise of security!

Routing Gone Bad—Example 1

Where should I send traffic with the destination address 10.1.1.1?







Routing Protocol Security

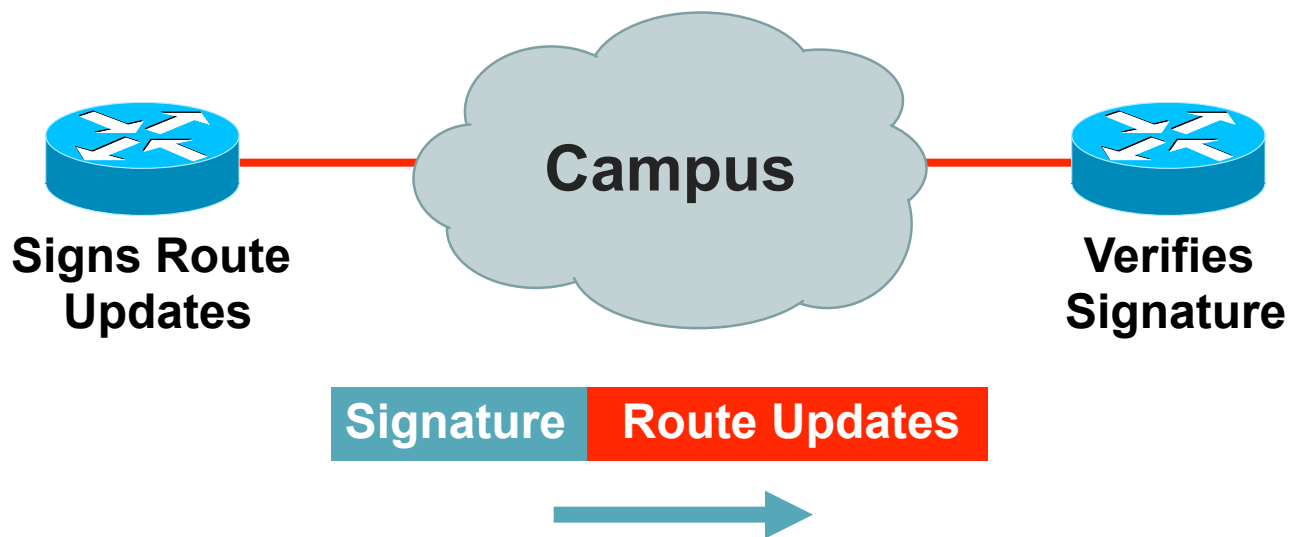
Routing Protocols Can Be Attacked

- Denial of service
- Smokescreens
- False information
- Reroute packets

May Be Accidental or Intentional

Secure Routing—Route Authentication

Configure Routing Authentication



Certifies **Authenticity** of Neighbor and **Integrity** of Route Updates



Route Authentication

- Shared key included in routing updates
 - Plain text—protects against accidental problems only
 - Message Digest 5 (MD5)—protects against accidental and intentional problems
- Multiple keys supported
- Supported for BGP, IS-IS, OSPF, RIPv2, and EIGRP
- Update keys before protocol timeout to avoid session bounce
- Often non-implemented
 - “Never seen an attack”
 - “My peer doesn’t use it”



Routing Protocol Security

- The case for using routing protocol authentication:
 - RIP is not the only protocol susceptible to this attack
 - OSPF, IS-IS, and BGP are all trusting by nature
 - Implementations of routing protocols are widely available for many operating systems
 - Easy to install on a compromised host
 - Easy to implement by a disgruntled employee
 - Easy to install by a well-meaning—but clueless—insider
 - Secure your routing protocols such that only authenticated neighbors and peers can participate
 - Limit your usage of routing protocols to only the links that must be included



Agenda: Securing RIP

- Routing Protocol Authentication
 - Securing RIP
- Securing OSPF
- Securing IS-IS
- Securing BGP
- Additional Routing Security



RIP Authentication

- RIP authentication only available when using RIPv2
 - Three types of authentication are available:
 - None (default)
 - Simple (clear text)
 - MD5
 - Using MD5 authentication, a trailer is added to each RIPv2 packet
 - Contains a keyed MD5 hash of the packet contents and a shared key
 - Provides *integrity*, but not *availability* or *confidentiality*
 - Two places to configure RIPv2 authentication:
 - Globally, for all RIPv2 neighbors (interfaces)
 - On an individual interface (neighbor level)



RIP Configuration (Global)

```
[edit protocols rip]
lab@R1# show
authentication-type md5;
authentication-key "$9$fQ6AEhr1vL1RhrvM-dqmfz9pKvL7dsO1";
# SECRET-DATA
group inside {
    neighbor fe-0/0/1.0;
}
group outside {
    neighbor fe-0/0/0.0;
}
```

In this example, MD5 authentication is enabled for all interfaces (neighbors) running RIPv2



RIP Configuration (Neighbor)

```
[edit protocols rip]
lab@R1# show
group inside {
    neighbor fe-0/0/1.0 {
        authentication-type md5;
        authentication-key "$9$qPz6B1hcrvu01hrlXxjHqf39yrv8xdtu";
# SECRET-DATA
    }
}
group outside {
    neighbor fe-0/0/0.0;
}
```

In this example, MD5 authentication is enabled for a single interface running RIPv2



Agenda: Securing OSPF

- Routing Protocol Authentication
- Securing RIP
- Securing OSPF
- Securing IS-IS
- Securing BGP
- Additional Routing Security

OSPF Authentication, Configuration

- Authentication occurs within an individual area
 - Three types are supported: none, simple, and MD5
- Each interface requires an authentication key
 - Multiple interfaces can use the same key
 - Keys are always encrypted in the configuration
- By default, the authentication type is set to none
 - Effectively means no authentication is performed
- Type simple uses a plain-text password

```
[edit protocols ospf]
```

```
user@host# show
```

```
area 0.0.0.2 {
```

```
    authentication-type simple;
```

```
    interface ge-0/0/0.0 {
```

```
        authentication-key "$9$-TbwgPfzn6A";
```

```
    }
```




MD5 Authentication Configuration

- Using MD5 authentication, a message digest is generated and appended to the end of each OSPF packet
 - Contains a keyed MD5 hash of the packet contents and a shared key
 - Provides *integrity*, but not *availability* or *confidentiality*
- Each interface requires an authentication key
 - Multiple interfaces can use the same key
 - Keys are always encrypted in the configuration
- Each key requires a key ID value ranging from 0 to 255
 - If omitted, a value of 0 is used

```
[edit protocols ospf]
user@host# show
area 0.0.0.1 {
    authentication-type md5;
    interface so-0/3/1.0 {
        authentication-key "$9$u18b0IcyrvL7VKM" key-id 10;
    }
}
```



Agenda: Securing IS-IS

- Routing Protocol Authentication
- Securing RIP
- Securing OSPF
- Securing IS-IS
- Securing BGP
- Additional Routing Security



IS-IS Authentication

- Authentication can occur within multiple places
 - Level 1
 - Level 2
 - Interface
- Three authentication types are supported
 - None (default)
 - Simple
 - MD5
- Using HMAC-MD5 authentication, TLV 10 is included in each IS-IS PDU
 - TLV contains an HMAC-MD5 hash of the packet contents and a shared key

Provides *integrity*, but not *availability* or *confidentiality*



Authentication Configuration

- Level authentication affects all IS-IS PDUs
 - Link-state, sequence number, and hello
- Per-interface authentication takes precedence over per-level settings

```
[edit protocols isis]
user@host# show
level 1 {
    authentication-key "$9$bssYomPQ69pkq39puhc8X7V2a"; # SECRET-DATA
    authentication-type md5;
}
level 2 {
    authentication-key "$9$dXVYoDjqQ39gomTz6CAvW8X-ViHmFnCDi1h"; # SECRET-DATA
    authentication-type simple;
}
interface fe-0/0/0.0 {
    level 2 {
        hello-authentication-key "$9$1sEEclw4JH-d2oGq.Ctp01h7NbgaU"; # SECRET-DATA
        hello-authentication-type md5;
```



Authentication Issues

- Hello authentication only secures IS-IS hello packets
 - Determines whether an adjacency forms between two routers
- Level 1 or Level 2 authentication can be disabled for specific PDUs
 - LSP packets
 - CSNP packets (`no-csnp-authentication`)
 - PSP packets (`no-psnp-authentication`)
 - IS-IS hello packets (`no-hello-authentication`)
- Authentication for LSPs allows other routers to read the TLV values and use that information in the SPF calculation
- Disables the authentication check with the `no-authentication-check` command
 - Useful for migration purposes

OSPF and ISIS Authentication Example

OSPF

- interface ethernet1
- ip address 10.1.1.1
255.255.255.0
- ip ospf message-digest-
key 100 md5 qa*>HH3
- !
- router ospf 1
- network 10.1.1.0
0.0.0.255 area 0
- area 0 authentication
message-digest

ISIS

- interface ethernet0
- ip address 10.1.1.1
255.255.255.0
- ip router isis
- isis password pe#
\$rt@s level-2



Agenda: Securing BGP

- Routing Protocol Authentication
- Securing RIP
- Securing OSPF
- Securing IS-IS
- ➔ Securing BGP
- Additional Routing Security



BGP Authentication

- BGP authentication:
 - Two types of authentication available
 - None (default)
 - MD5
 - Using MD5 authentication, an extension is included in each TCP segment
 - Contains a 16-byte MD5 hash of the packet contents and a shared key
 - Provides *integrity*, but not *availability* or *confidentiality*
- Three places to apply BGP authentication:
 - Globally for all peers
 - For all peers in an individual group
 - For a single peer



Configuration—Global, Group, or Peer

```
[edit protocols bgp]
lab@R1# show
authentication-key "$9$9tiTtpByrvMLNhSrvLXwsfTz"; # SECRET-DATA
group external {
    type external;
    neighbor 172.16.1.2 {
        authentication-key "$9$JAUDkQz6/A0fTz6AtREVwY"; # SECRET-DATA
        peer-as 64513;
    }
    neighbor 172.16.1.3 {
        peer-as 64512;
    }
}
group internal {
    type internal;
    local-address 10.1.255.1;
    authentication-key "$9$.f5FtpB1Ey9ApBEhvMJGD"; # SECRET-DATA
    neighbor 10.1.255.2;
    neighbor 10.1.255.3;
```



BGP Route Authentication

```
router bgp 200
  no synchronization
  neighbor 4.1.2.1 remote-as 300
  neighbor 4.1.2.1 description Link to
    Excalibur
  neighbor 4.1.2.1 send-community
  neighbor 4.1.2.1 version 4
  neighbor 4.1.2.1 soft-reconfiguration
    inbound
  neighbor 4.1.2.1 route-map Community1 out
  neighbor 4.1.2.1 password 7 q23dc%$#ert
```



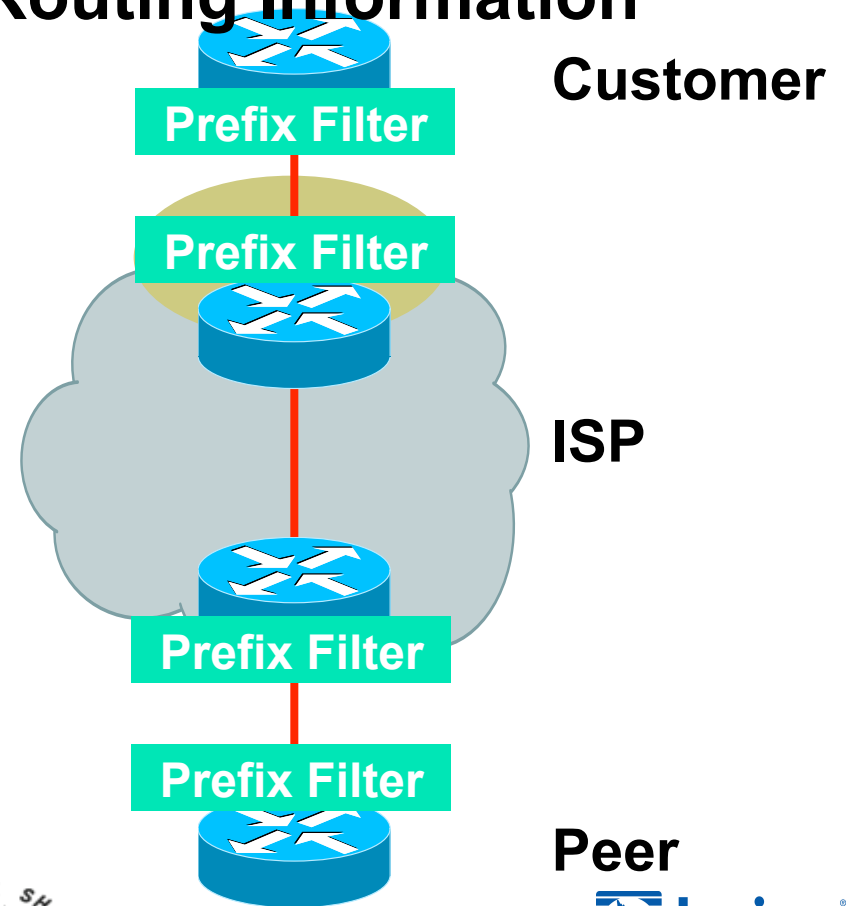
BGP Route Authentication

- Works per neighbor or for an entire peer-group
- Two routers with password mismatch:
 - %TCP-6-BADAUTH: Invalid MD5 digest from [peer's IP address]:11004 to [local router's IP address]:179
- One router has a password and the other does not:
 - %TCP-6-BADAUTH: No MD5 digest from [peer's IP address]:11003 to [local router's IP address]:179

Prefix Filters

Apply Prefix Filters to All eBGP Neighbors to Prevent Injection of False Routing Information

- To/from customers
- To/from peers
- To/from upstreams





Route Filtering

- Consider filtering routes on input to a protocol
 - You can do this on both RIP and BGP
 - The architecture of OSPF and IS-IS does not allow input filters to be placed on them
- Possible inbound filters include:
 - Reject any of your own routes advertised to you
 - Reject any external RFC 1918 routes advertised to you
 - Reject any external routes from reserved or unallocated address space
 - Reject any instance of the default route sent to you



BGP thoughts

- Consider what routes you should get from peer
 - Accept those
 - Filter all as default
- Customer (non transit) BGP routes should be received with
 - Customer AS Number, AS-Path length = 1,
 - Customer address range only
 - What degree of sub-netting will you allow
 - What is useful?

Extended ACL for a BGP Distribute List

```
access-list 150 deny ip host 0.0.0.0 any
access-list 150 deny ip 10.0.0.0 0.255.255.255 255.0.0.0 0.255.255.255
access-list 150 deny ip 127.0.0.0 0.255.255.255 255.0.0.0 0.255.255.255
access-list 150 deny ip 169.254.0.0 0.0.255.255 255.255.0.0 0.0.255.255
access-list 150 deny ip 172.16.0.0 0.15.255.255 255.240.0.0 0.15.255.255
access-list 150 deny ip 192.0.2.0 0.0.0.255 255.255.255.0 0.0.0.255
access-list 150 deny ip 192.168.0.0 0.0.255.255 255.255.0.0 0.0.255.255
access-list 150 deny ip 224.0.0.0 31.255.255.255 224.0.0.0 31.255.255.255
access-list 150 permit ip any any
```

BGP with Distribute List Route Filtering

```
router bgp 65535
no synchronization
bgp dampening
neighbor 220.220.4.1 remote-as 210
neighbor 220.220.4.1 version 4
neighbor 220.220.4.1 distribute-list 150 in
neighbor 220.220.4.1 distribute-list 150 out
neighbor 222.222.8.1 remote-as 220
neighbor 222.222.8.1 version 4
neighbor 222.222.8.1 distribute-list 150 in
neighbor 222.222.8.1 distribute-list 150 out
no auto-summary
!
```




Prefix-List for a BGP Prefix List

```
ip prefix-list rfc1918-dsua seq 5 deny 0.0.0.0/8 le 32
ip prefix-list rfc1918-dsua seq 10 deny 10.0.0.0/8 le
32
ip prefix-list rfc1918-dsua seq 15 deny 127.0.0.0/8 le
32
ip prefix-list rfc1918-dsua seq 20 deny 169.254.0.0/16
le 32
ip prefix-list rfc1918-dsua seq 25 deny 172.16.0.0/12
le 32
ip prefix-list rfc1918-dsua seq 30 deny 192.0.2.0.0/24
le 32
ip prefix-list rfc1918-dsua seq 35 deny 192.168.0.0/16
le 32
ip prefix-list rfc1918-dsua seq 40 deny 224.0.0.0/3 le
32
ip prefix-list rfc1918-dsua seq 45 permit 0.0.0.0/0 le 32
```



BGP with Prefix-List Route Filtering

```
router bgp 65535
no synchronization
bgp dampening
neighbor 220.220.4.1 remote-as 210
neighbor 220.220.4.1 version 4
neighbor 220.220.4.1 prefix-list rfc1918-dsua in
neighbor 220.220.4.1 prefix-list rfc1918-dsua out
neighbor 222.222.8.1 remote-as 220
neighbor 222.222.8.1 version 4
neighbor 222.222.8.1 prefix-list rfc1918-dsua in
neighbor 222.222.8.1 prefix-list rfc1918-dsua out
no auto-summary
!
```



Agenda: Additional Routing Security

- Routing Protocol Authentication
- Securing RIP
- Securing OSPF
- Securing IS-IS
- Securing BGP
- ➔ Additional Routing Security



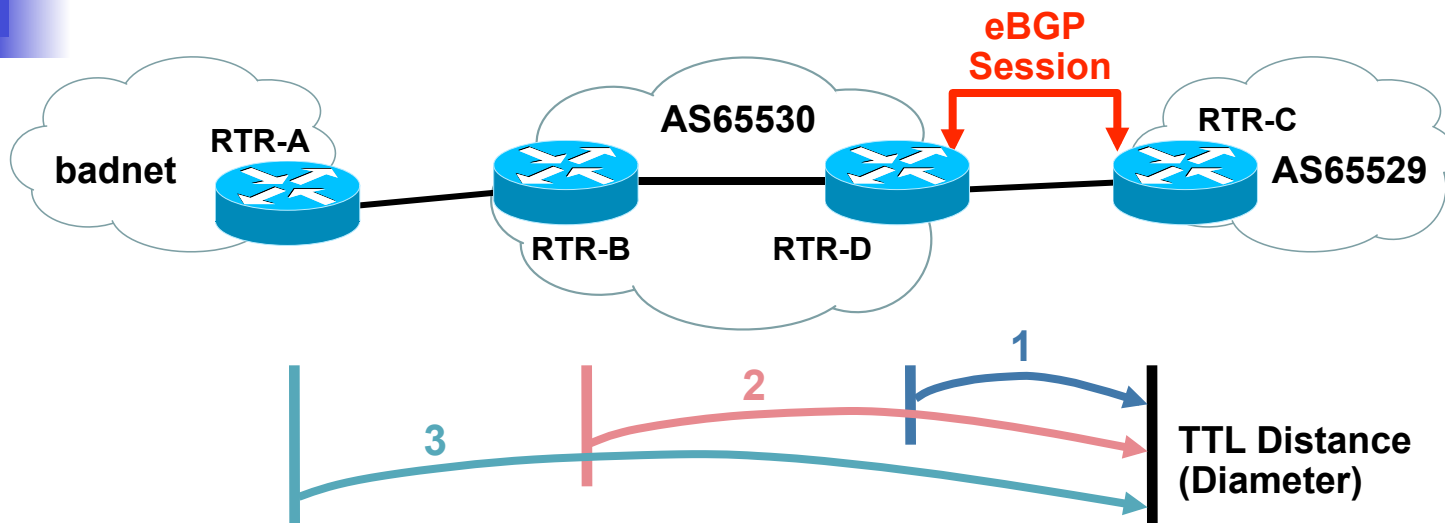
Other BGP thoughts

- Clean routes
 - Will you accept MEDS?
 - Leave communities alone
 - If you honor extended community format all should be OK
 - AS:nn
 - Is default allowed
 - Filter martians and bogons

BGP Support for TTL Security Check

- AKA BGP TTL Security Hack (BTSH)
- Protects eBGP sessions from CPU attacks using forged IP packets
- Prevents attempts to hijack eBGP session by attacker not part of either BGP network or that is not between the eBGP peers
- Configure minimum Time To Live (TTL) for incoming IP packets from a specific eBGP peer
 - BGP session established and maintained only if TTL in IP packet header is equal to or greater than configured TTL value. Initial TTL set to 255
 - If value is less than configured value packet is silently discarded and no ICMP message generated
- Not supported for iBGP and occurs after MD5 check if enabled
- Available in 12.0(27)S, 12.3(7)T, and 12.2(25)S
 - http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123newft/123t/123t_7/gt_btsh.htm

BGP TTL Security Check: How Does It Work?



Example on RTR-C:

```
router bgp 65529
neighbor 10.1.1.1 ttl-security hops 1
! expected TTL value in the IP packet header is 254
```

- Spoofed IP packets may have correct IP source and destination addresses (and TCP source and destination ports); however, unless these packets originate on a network segment that is between the eBGP peers, the TTL values will be less than the "minimum" configured in the BGP TTL security check



Not really security related...but...

- Extensive use of policies to provide customer services
 - E.g. Provider provisioned Local Pref
 - Examples at www.sprint.net
 - Under BGP policies



Martian Addresses

- One way to filter is to add prefixes to your martian address list
 - Address prefixes for which the routers ignore all associated routing information
- Martians are not installed into the routing table
- In JUNOS software, the default martian addresses are:
 - 0.0.0.0/8 orlonger
 - 127.0.0.0/8 orlonger
 - 128.0.0.0/16 orlonger
 - 191.255.0.0/16 orlonger
 - 192.0.0.0/24 orlonger
 - 223.255.255.0/24 orlonger
 - 240.0.0.0/4 orlonger

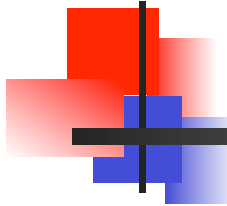


Adding Martian Addresses

- Additional prefixes can be added to the martian list
 - This example adds all RFC 1918 addresses to the list
- Configured at the `routing-options` hierarchy level

```
routing-options {  
    martians {  
        destination-prefix match-type;  
    }  
}
```

```
[edit]  
routing-options {  
    martians {  
        10.0.0.0/8 orlonger;  
        172.16.0.0/12 orlonger;  
        192.168.0.0/16 orlonger;
```



Bogons

- Addresses that are unallocated
- MUST MUST MUST MUST MUST MUST
 - Keep them up to date!
- www.cymru.com



Additional BGP Security

- IPsec transport mode between peers
- Consider enabling route damping
 - Limits the number of times an unstable (or compromised) peer can destabilize your own routing process
- Consider establishing a limit to the number of prefixes you will accept from a peer

```
[edit protocols bgp group external]
lab@R1# show
neighbor 10.1.1.1 {
    family inet {
        any {
            prefix-limit {
                maximum 125000;
                teardown 85 idle-timeout 30;
            }
        }
    }
}
peer-as 64512;
}
```



CISCO

peer-as 64512;

}





IPSec—Global, Group, or Peer

```
[edit protocols bgp]
```

```
lab@R1# show
```

```
Ipsec-sa All-BGP-  
Neighbors
```

```
group external {  
    type external;  
    neighbor 172.16.1.2 {  
        ipsec-sa Just-1-  
eBGP-Neighbor  
        peer-as 64513;  
    }  
    neighbor 172.16.1.3 {  
        peer-as 64512;  
    }  
}
```

```
group internal {  
    type internal;  
    local-address 10.1.255.1;  
    ipsec-sa Just-iBGP-Neighbors  
    neighbor 10.1.255.2;  
    neighbor 10.1.255.3;  
}
```





More BGP Security

- Since JUNOS software Release 5.4, no response to unconfigured peers
 - Negates TCP DoS attacks against TCP port 179
 - Combined with firewall filters

```
[edit policy-options]
```

```
user@R1# show | display inheritance
```

```
prefix-list ibgp-peers {
```

```
##
```

```
## apply-path was expanded to:
```

```
##      10.2.255.2;
```

```
##      10.2.255.3;
```

```
##
```

```
apply-path "protocols bgp group <*> neighbor <*>";
```



Source MAC Address Filtering

In shared peering environments over broadcast capable media, consider using source MAC address filtering

```
[edit interfaces fe-0/0/3]
lab@R1# show
fastether-options {
    source-filtering;
    source-address-filter {
        00:e0:18:01:18:4c;
    }
}
unit 0 {
    family inet {
        address 10.2.100.1/24;
    }
}
```

Verifying Authentication

- Authentication information available with the `show ospf interface detail` command
 - Type of authentication is displayed
 - Key ID values shown if appropriate

```
user@host> show ospf interface detail
```

Interface	State	Area	DR ID	BDR ID	Nbrs
fe-0/0/2.0	DR	0.0.0.0	192.168.36.1	192.168.24.1	1

Type LAN, address 10.222.4.2, mask 255.255.255.0, MTU 1500, cost 1

DR addr 10.222.4.2, BDR addr 10.222.4.1, adj count 1, priority 128

Hello 10, Dead 40, ReXmit 5, Not Stub

Auth type MD5, Active key id 4, Start time 2003 Apr 14 11:05:00 UTC

fe-0/0/3.0	DRother	0.0.0.0	0.0.0.0	0.0.0.0	0
------------	---------	---------	---------	---------	---

Type LAN, address 1.1.1.2, mask 255.255.255.0, MTU 1500, cost 1

adj count 0, priority 128

Hello 10, Dead 40, ReXmit 5, Not Stub

Auth type Password





BGP Attack Vectors

- Understanding BGP Attack Vectors will help you plan and prioritize the techniques deployed to build greater resistance into the system.
- The following documents will help you gain perspective on the realistic Risk Assessment:
 - NANOG 25 - BGP Security Update
 - <http://www.nanog.org/mtg-0206/barry.html>
 - NANOG 28 - BGP Vulnerability Testing: Separating Fact from FUD
 - <http://www.nanog.org/mtg-0306/franz.html>
- Look for the *updates* links to get the latest risk assessments.
 - http://www.cisco.com/security_services/ciag/initiatives/research/projectsummary.html

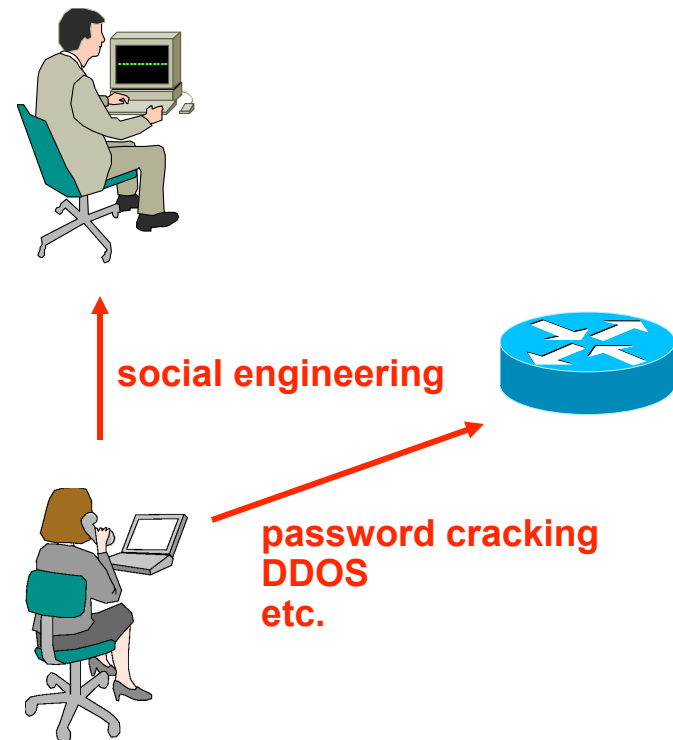


Whacking the BGP Session

- Four Macro Ways you can Whack the BGP Session:
 - Saturate the Receive Path Queues: BGP times out
 - Saturate the link: link protocols time out
 - Drop the TCP session
 - Drop the IGP causing a recursive loop up failure

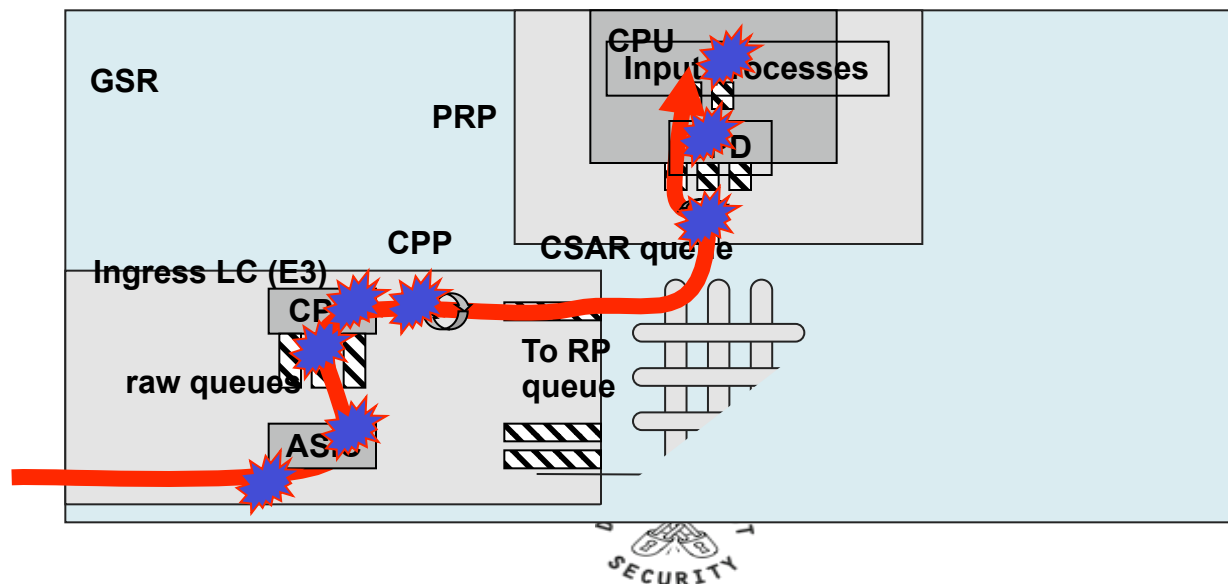
Attacking Routing Devices

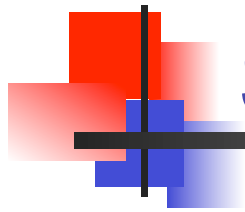
- All the normal host attack methods apply to routers
 - Social engineering
 - Password cracking
 - Denial of service
 - etc.
- What an attacker needs:
 - Access to the router
 - (or)
 - Access to the network



Saturate the Receive Path Queues

- Routers usually have various *receive path* queues that are hit as the packet heads for the TCP Stack.
- Saturation Attacks fill these queues – knocking out valid packets from the queues.
- Consequence: BGP Times out – Dropping the BGP Session





Saturate the Link

- DOS Attacks Saturating the link will knock out valid control plane packets.
- Link packet over POS, ATM, or Ethernet will drop out – which drop out the link – which drop out the FIB's next hop – which knocks out the BGP Entries
- This is a very effective brute force attack.

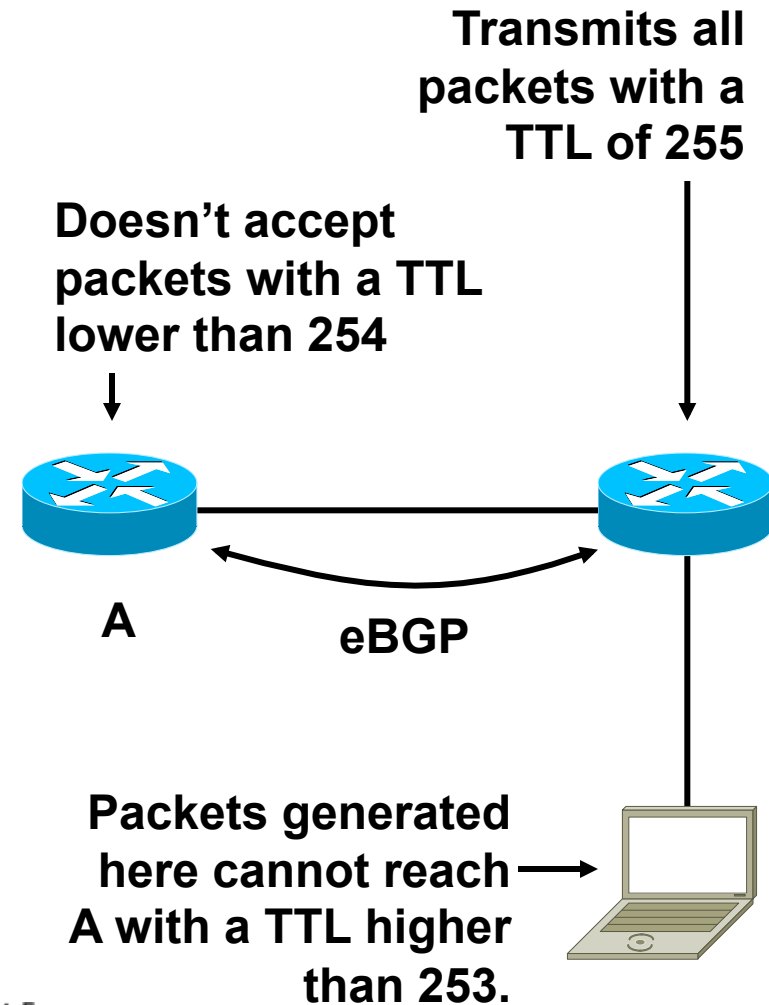


Drop the TCP Session

- Dropping the TCP Session was thought to require a breath of packets.
- TCP Session can be dropped with a RST or a SYN (per RFC).
- Successful L4 Spoof is required
 - Match source address
 - Match source port
 - Match destination address (obvious)
 - Match destination port
 - Match Sequence Number (now just get inside the window)

Generalized TTL Security Mechanism

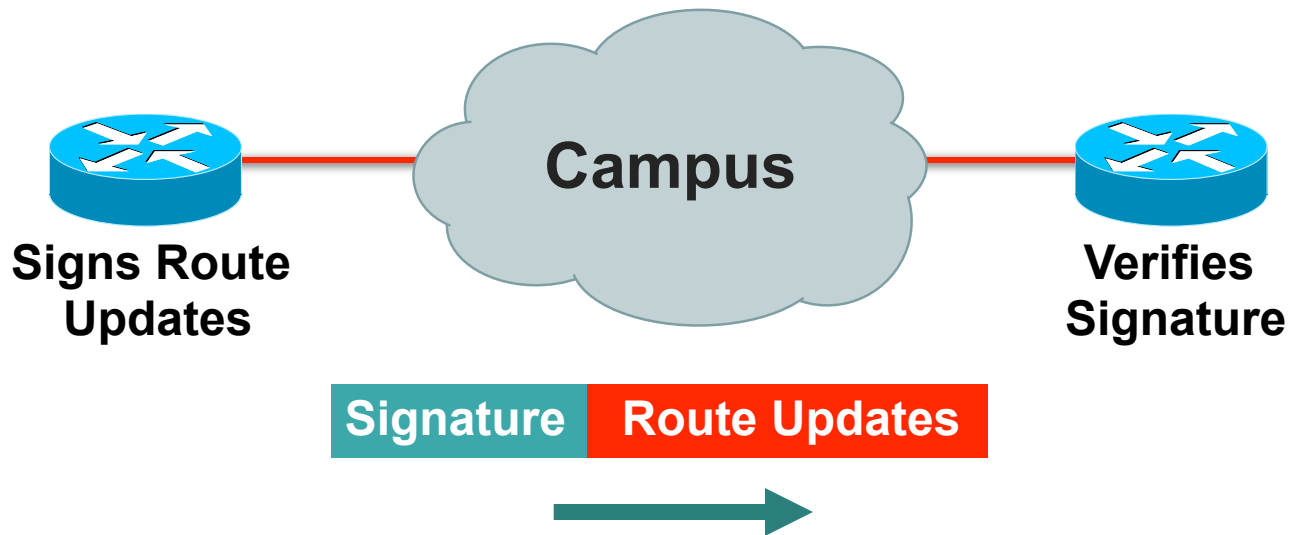
- GTSH is a hack which protects the BGP peers from multihop attacks.
- Routers are configured to transmit their packets with a TTL of 255, and to reject all packets with a TTL lower than 254 or 253.
- A device which isn't connected between the routers cannot generate packets which will be accepted by either one of them.



Secure Routing

Route Authentication

Configure Routing Authentication



Certifies **Authenticity** of Neighbor
and **Integrity** of Route Updates



Peer Authentication

- MD5 Peer authentication can protect against:
 - Malformed packets tearing down a peering session
 - Unauthorized devices transmitting routing information
- MD5 Peer authentication cannot protect against:
 - Reset routing protocol sessions due to denial of service attacks
 - Incorrect routing information being injected by a valid device which has been compromised



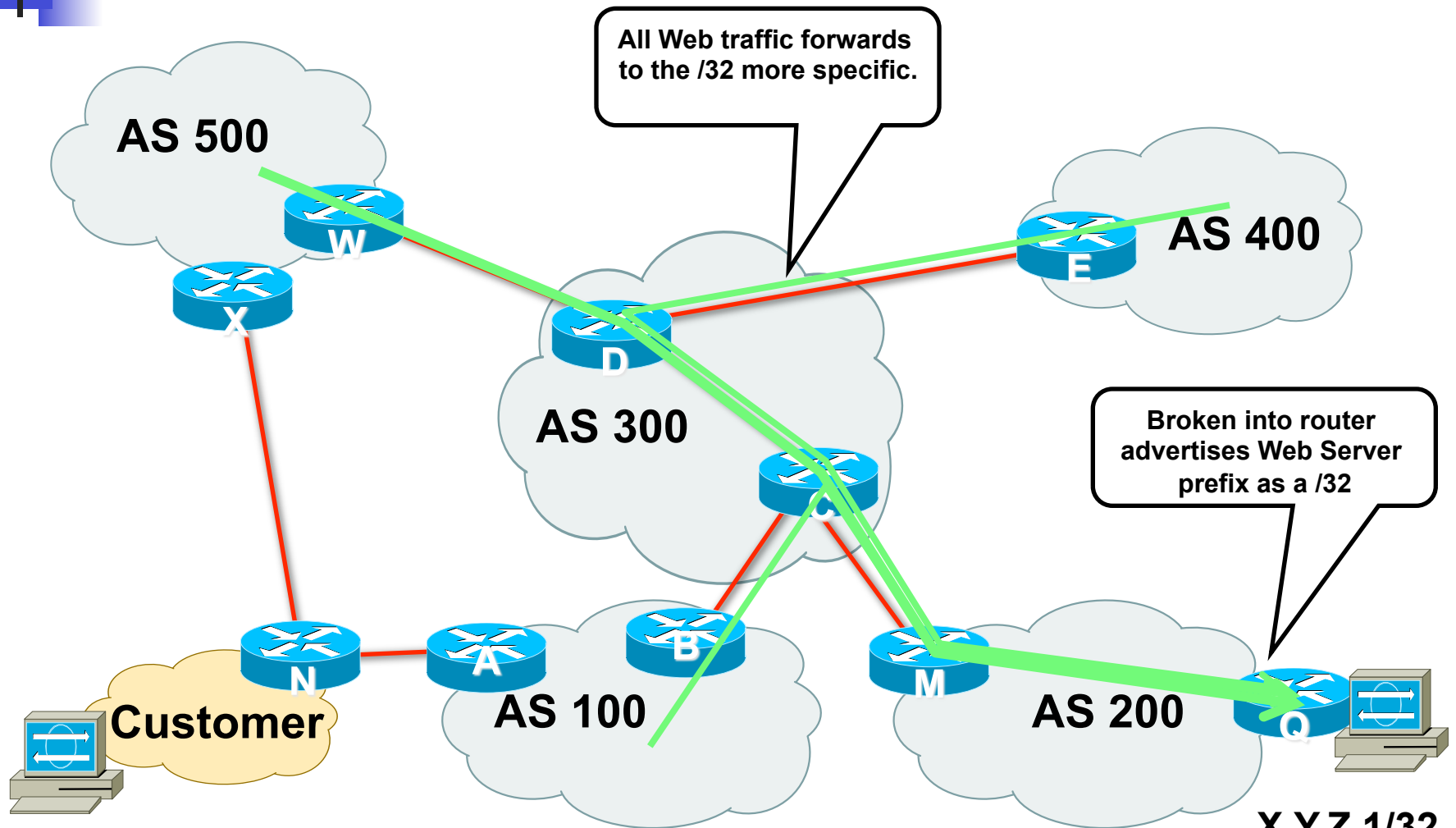
Drop the IGP

- Miscreant Success Principle - If you cannot take out the target, move the attack to a coupled dependency of the target.
- BGP's coupled dependency is the IGP it requires for recursive look-up.
- EIGRP and OSPF are both open to external attacks.

- 



What is a prefix hijack?

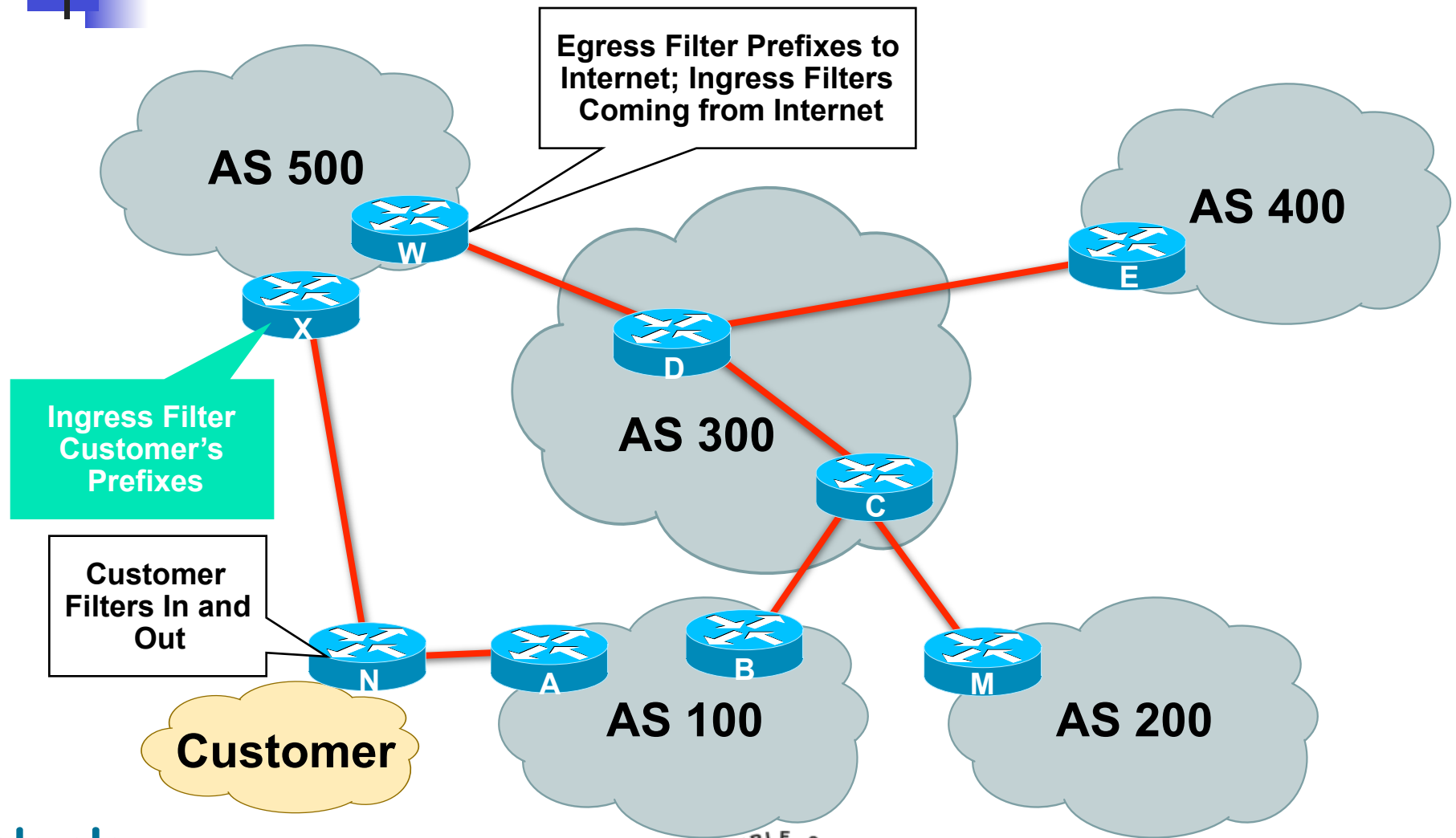


Malicious Route Injection

What can ISPs Do?

- Customer Ingress Prefix Filtering!
- ISPs should only accept customer prefixes which have been assigned or allocated to their downstream customers.
- For example
 - Downstream customer has 220.50.0.0/20 block.
 - Customer should only announce this to peers.
 - Upstream peers should only accept this prefix.

Where to Prefix Filter?





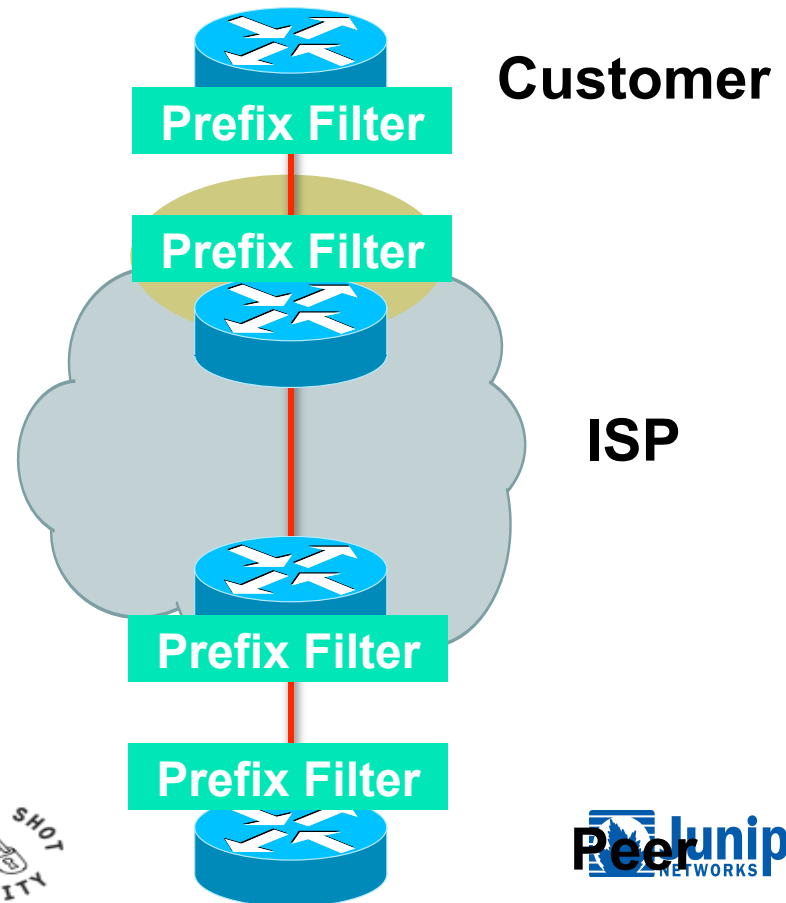
Bogons and Special Use Addresses

- IANA has reserved several blocks of IPv4 that have yet to be allocated to a RIR:
 - <http://www.iana.org/assignments/ipv4-address-space>
- These blocks of IPv4 addresses should never be advertised into the global internet route table
- Filters should be applied on the AS border for all inbound and outbound advertisements
- Special Use Addresses (SUA) are reserved for special use :-)
 - Defined in RFC3330
 - Examples: 127.0.0.1, 192.0.2.0/24

Prefix Filters: Application

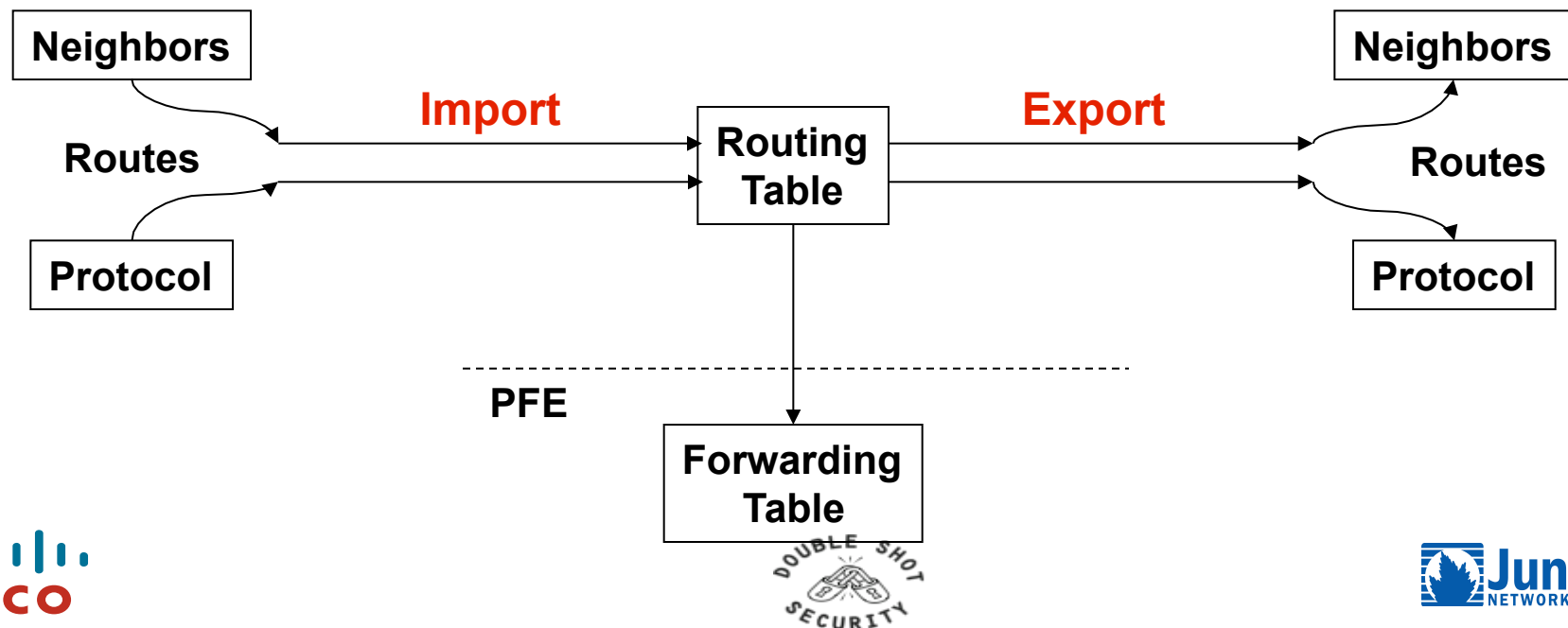
Apply Prefix Filters to All eBGP Neighbors

- To/from customers
- To/from peers
- To/from upstreams



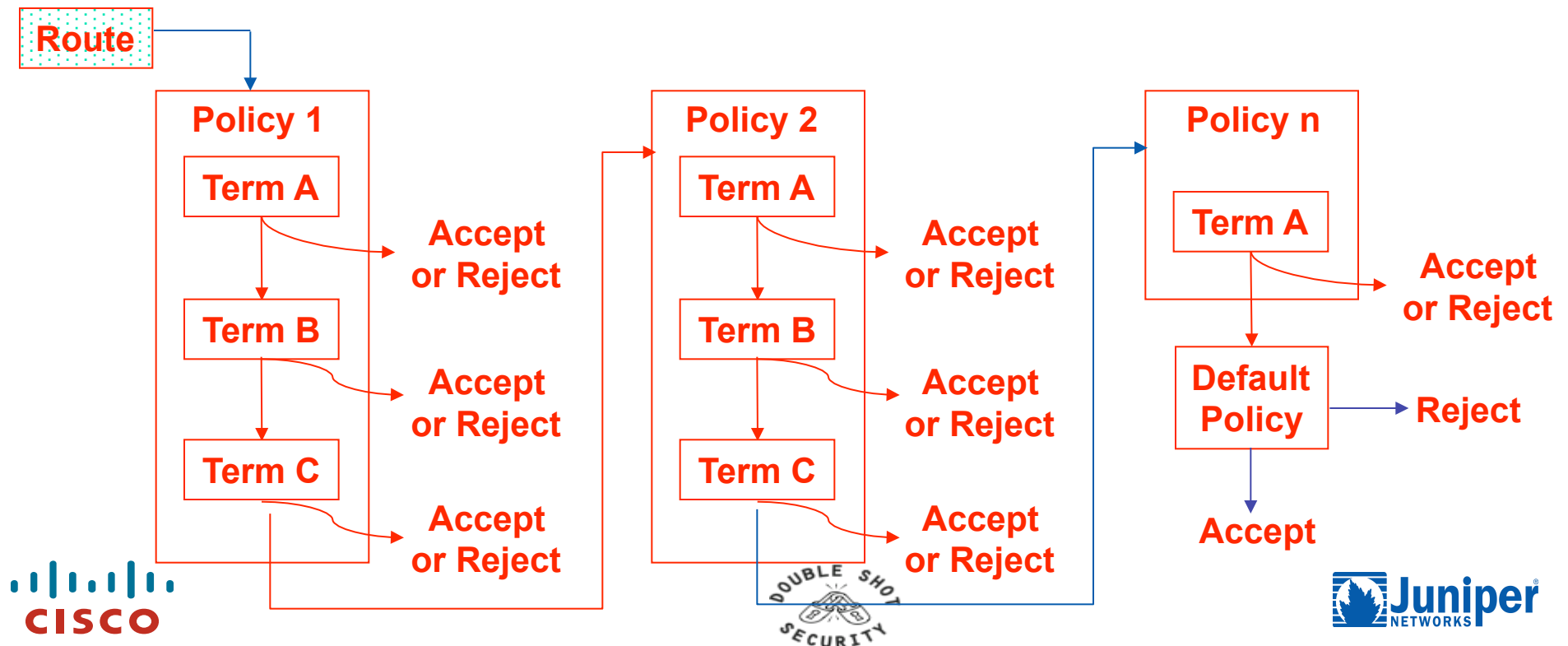
Import and Export Policies

- Perform policy filtering with respect to the software routing table



Routing Policy Flow

- Policies can be chained together
- Evaluation normally proceeds left to right until a *terminating* action is reached
 - Terminating actions are accept or reject
- Individual policies can contain a collection of terms
 - Flow control actions such as `next-policy` supported





Generic Policy Syntax

Basic policy syntax:

```
policy-options {  
  policy-statement policy-name {  
    term term-name {  
      from {  
        match-conditions;  
      }  
      then {  
        action;  
      }  
    }  
  }  
}
```

**A policy
can have
multiple
terms**



Match Conditions

- Policies typically contain some form of match criterion
- Possibilities include:
 - Neighbor address
 - Protocol (source of information)
 - BGP, direct, DVMRP, IS-IS, local, MPLS, OSPF, PIM, RIP, static, aggregate
 - Routing protocol information
 - OSPF area ID
 - IS-IS level number
 - BGP attributes
 - Regular expression-based matches for AS path and communities



Match Actions

- The action associated with a given term/policy is performed for matching routes:
 - Terminating actions
 - Accept route
 - Reject (or suppress) route
 - Flow control actions
 - Skip to next policy
 - Skip to next term
 - Modify attributes actions
 - Metric
 - Preference
 - Color
 - Next-hop address



Default Policies

- Every protocol has a default policy
 - The default policy is applied implicitly at the end of the policy chain; can be overridden with `default-action` statement
- IS-IS and OSPF
 - Import: Accept all routes learned from that protocol
 - Technically, accept all LSPs/LSAs flooded by that protocol
 - Export: Reject everything
 - LSP/LSA flooding announces (IS-IS/OSPF) learned and local routes
- RIP
 - Import all learned RIP routes, export nothing
 - RIP requires export policy to announce RIP (or other) routes
- BGP
 - Import all routes learned from BGP neighbors
 - Export all active routes learned from BGP neighbors to all BGP neighbors
 - EBGp-learned routes are exported to all BGP peers
 - IBGP-learned routes are exported to all EBGp peers (assumes logical IBGP full mesh)



A Policy Example

- Write a policy statement at the `[edit policy-options]` hierarchy:

```
[edit policy-options]
user@host# show policy-statement advertise-ospf
term pick-ospf {
    from protocol ospf;
    then accept;
}
```

- Apply the policy to one or more routing protocol in the `import`, `export`, or both directions:

```
[edit protocols bgp]
user@host# set export advertise-ospf
```



Another Policy Example

Specifying multiple conditions in a `from` statement means that *all* criteria must match before the action is taken

```
[edit]
user@host# show policy-options
policy-statement isis-level2 {
  term find-level2-routes {
    from {
      protocol isis;
      level 2;
    }
    then accept;
  }
}
```

Logical AND Function

Applying Policy

- You must apply policies before they can take effect
- Link-state protocols (IS-IS and OSPF) have only export filtering points
- BGP and RIP support both import and export policies

```
[edit protocols]
user@host# show
bgp {
    import bgp-import;
    export bgp-export;
}
ospf {
    export ospf-export;
}
```




Apply Routing Policy to BGP

- BGP has three filtering points per direction:
 - Global
 - Groups of neighbors
 - Individual neighbors
- Only the *most* specific policies are applied to a particular peer
 - Neighbor policy overrides group and global policies
 - Group policy overrides global policy

BGP Policy Application Example

```
[edit protocols]
user@host# show
bgp {
  export local-customers;
  group meganet-inc {
    type external;
    import [ martian-filter long-prefix-filter as-47-filter ];
    peer-as 47;
    neighbor 1.2.2.4;
    neighbor 1.2.2.5;
  }
  group problem-child {
    type external;
    import [ as-47-filter long-prefix-filter martian-filter ];
    export kill-private-addresses;
    peer-as 54;
    neighbor 1.2.2.6;
    neighbor 1.2.2.7;
    neighbor 1.2.2.8 {
      import [ reject-unwanted as-666-routes ];
    }
  }
}
```



Route Filters

- Use route filters to match an individual route (or groups of routes)
 - You can specify multiple route filters within a single term
 - General syntax in the form of:

route-filter prefix/prefix-length match-type actions;

- Route filter evaluation has special rules according to the match type
 - Match types specify different sets of routes:
 - exact
 - orlonger
 - longer
 - upto
 - through
 - prefix-length-range
 - Policy `test` function is useful for route-filter debugging



Route Filter Match Types (2 of 5)

- `orlonger`
 - Match the specified prefix and mask exactly
 - Also match any routes that start with the same prefix and have longer masks

from route-filter 192.168/16 `orlonger`;

	200.1.1.0/24		192.168.0.0/24
	192.0.0.0/8		192.168.1.0/24
	192.168.0.0/16		192.168.64.0/25
	192.168.1.0/16		192.168.32.0/26
	192.168.0.0/17		192.168.192.0/26
	192.168.128.0/17		192.168.1.1/32



Route Filter Match Types (3 of 5)

- longer
 - Do *not* match the specified prefix and mask exactly
 - Match only the routes that start with the same prefix and have longer masks
- from route-filter 192.168/16 longer;

	200.1.1.0/24		192.168.0.0/24
	192.0.0.0/8		192.168.1.0/24
	192.168.0.0/16		192.168.64.0/25
	192.168.1.0/16		192.168.32.0/26
	192.168.0.0/17		192.168.192.0/26
	192.168.128.0/17		192.168.1.1/32



Route Filter Match Types (4 of 5)

- upto
 - Match the specified prefix and mask exactly
 - Also match any routes that start with the same prefix and have a mask no longer than the second value specified

```
from route-filter 192.168/16 upto /24;
```

	200.1.1.0/24		192.168.0.0/24
	192.0.0.0/8		192.168.1.0/24
	192.168.0.0/16		192.168.64.0/25
	192.168.1.0/16		192.168.32.0/26
	192.168.0.0/17		192.168.192.0/26
	192.168.128.0/17		192.168.1.1/32



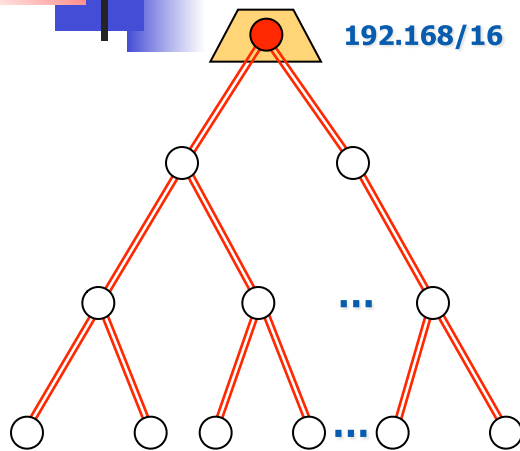
Route Filter Match Types (5 of 5)

- `prefix-length-range`
 - Match only routes that start with the same prefix and have a mask between the two values specified (inclusive match)

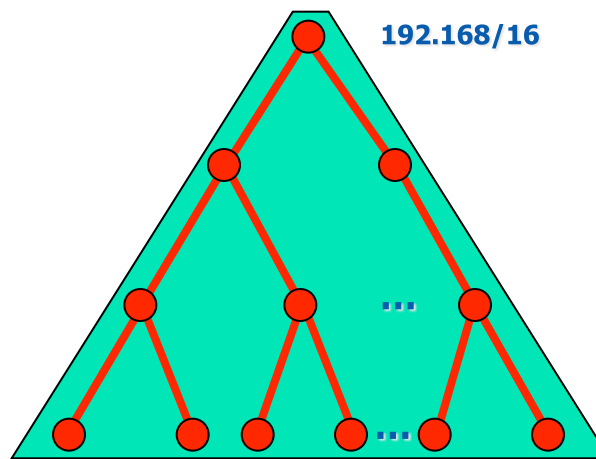
```
from route-filter 192.168/16 prefix-length-range /20-/24;
```

	200.1.1.0/24		192.168.0.0/22
	192.0.0.0/8		192.168.1.0/24
	192.168.0.0/16		192.168.64.0/25
	192.168.1.0/16		192.168.32.0/26
	192.168.0.0/17		192.168.192.0/26
	192.168.196.0/20		192.168.1.1/32

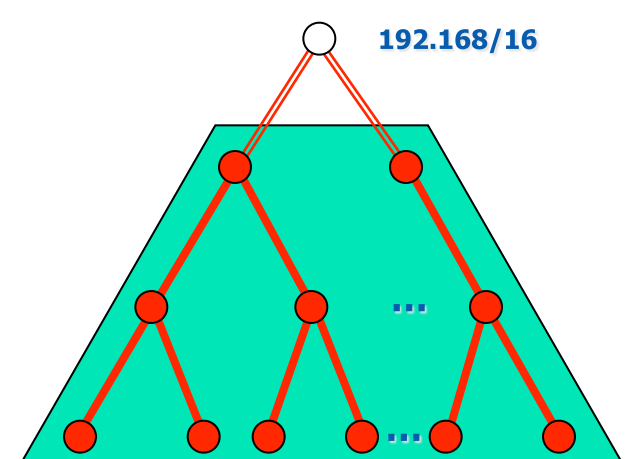
Match Types Summary



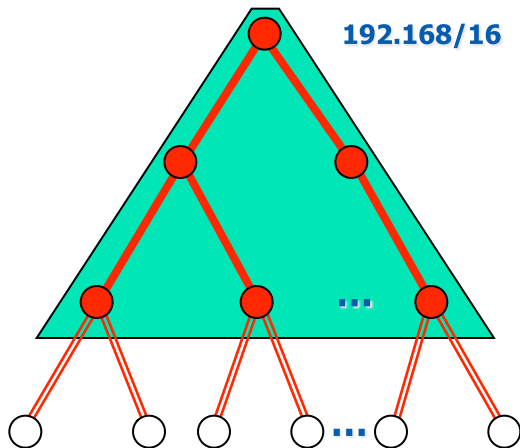
exact



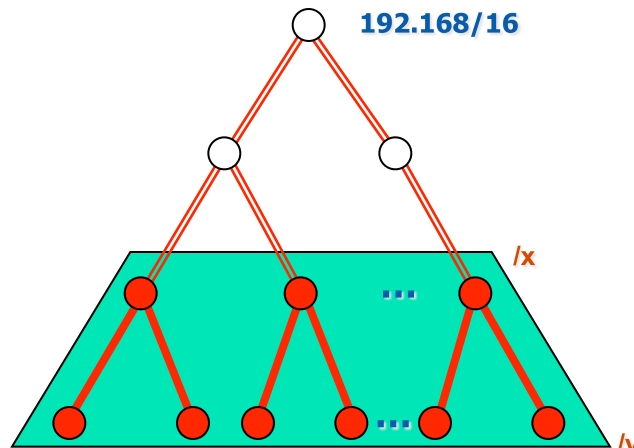
orlonger (down to /32)



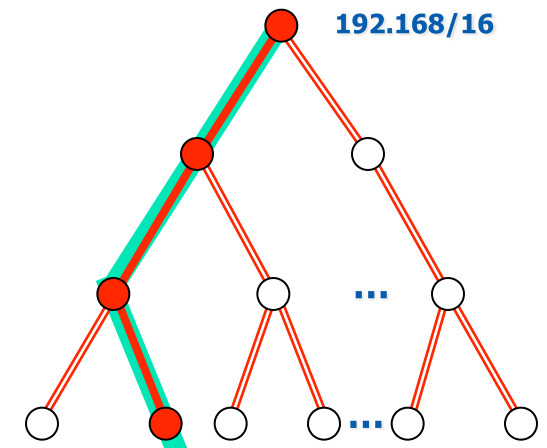
longer (down to /32)



upto



prefix-length-range /x-/y



through

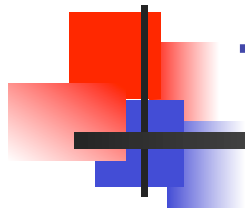


Route Filter Actions

```
term term-name {  
  from {  
    route-filter dest-prefix match-type actions;  
    route-filter dest-prefix match-type actions;  
  }  
  then actions;  
}
```

} Longest-
Match
Lookup

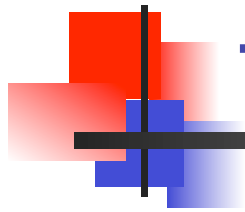
- Only one route filter in a given term can be considered a match
 - Longest-match lookup is performed on the prefix being evaluated
- If an action is specified to a route filter, it takes effect immediately
 - The global `then` portion of the term is ignored
 - If specific actions are not defined, the `then` portion of the term is executed for matching prefixes



Test Your Knowledge (1 of 2)

Which action is taken when this policy evaluates 10.0.67.43/32?

```
[edit policy-options policy-statement pop-quiz]
user@host# show
from {
    route-filter 10.0.0.0/16 orlonger accept;
    route-filter 10.0.67.0/24 orlonger;
    route-filter 10.0.0.0/8 orlonger reject;
}
then {
    metric 10;
    accept;
```

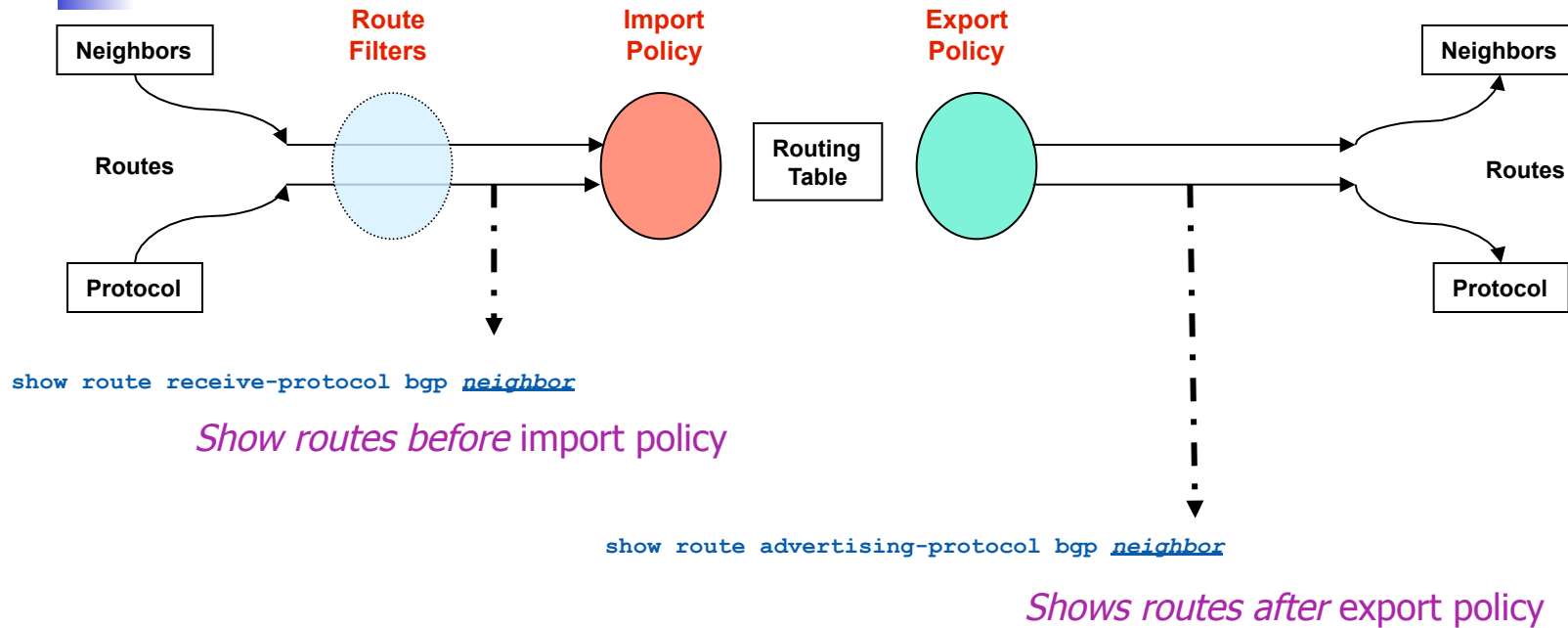


Test Your Knowledge (2 of 2)

Which action is taken when this policy evaluates 10.0.55.2/32?

```
[edit policy-options policy-statement pop-quiz]
user@host# show
from {
    route-filter 10.0.0.0/16 orlonger accept;
    route-filter 10.0.67.0/24 orlonger;
    route-filter 10.0.0.0/8 orlonger reject;
}
then {
    metric 10;
    accept;
```

Monitoring Policy Operation

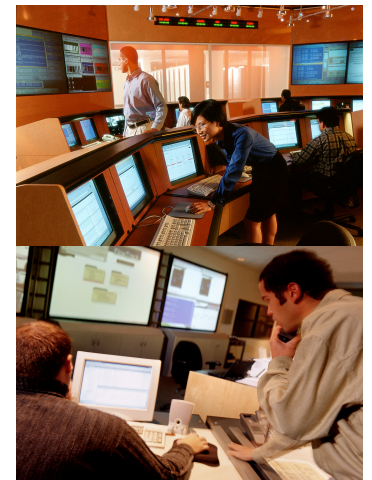
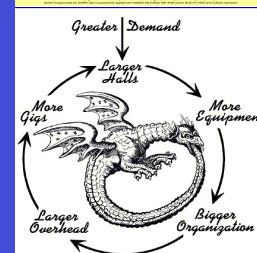
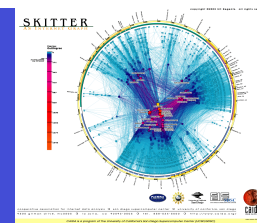


- The `show route receive-protocol` and `show route advertising-protocol` commands:



Lab: Securing routing protocols

Remote Trigger Black Hole [RTBH]

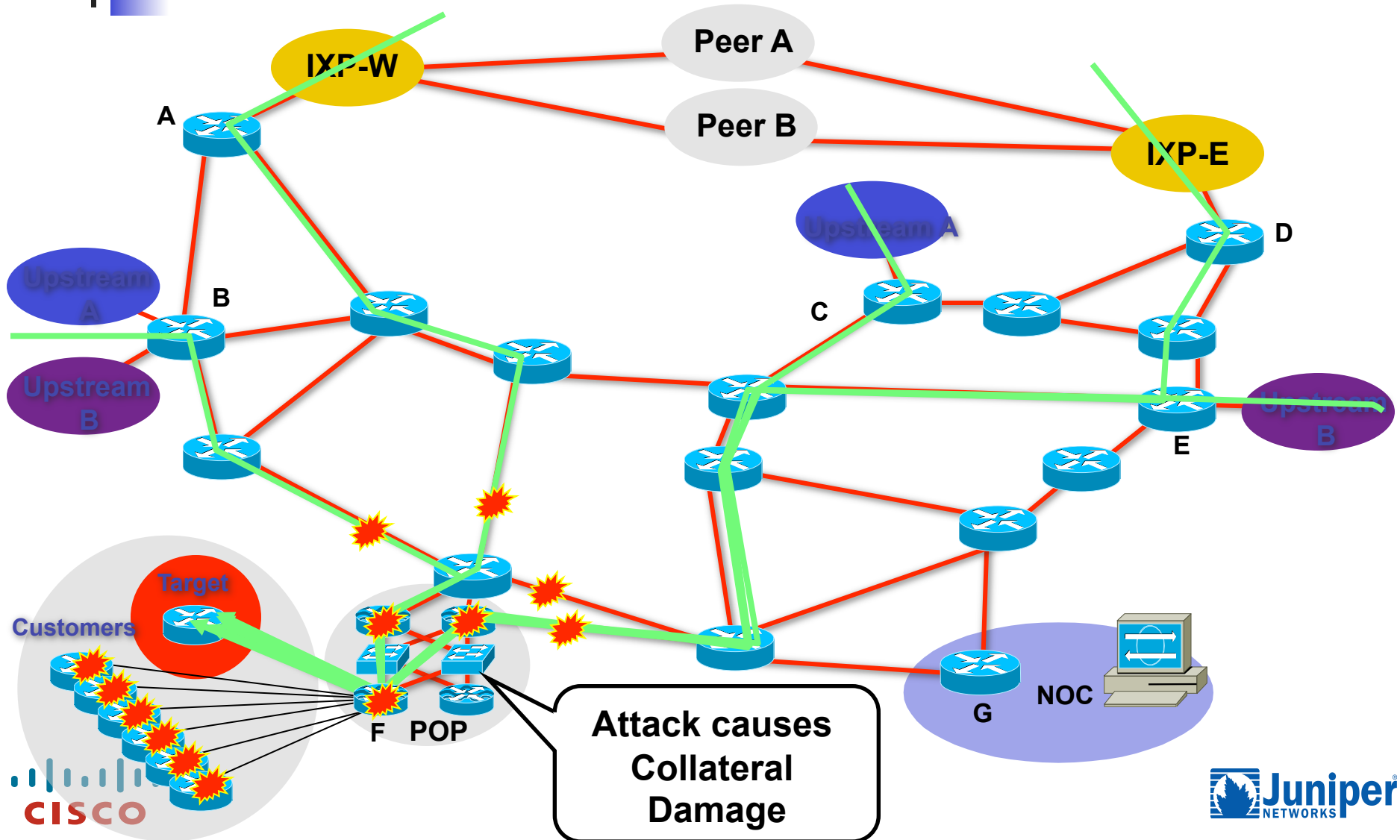




RTBH Filtering

- We use BGP to trigger a network wide response to a range of attack flows.
- A simple static route and BGP will allow an ISP to trigger network wide black holes as fast as iBGP can update the network.
- This provides ISPs a tool that can be used to respond to security related events or used for DOS/DDOS Backscatter Tracebacks.

Customer is DOSed – Before – Collateral Damage





Shunning with uRPF and BGP

- BGP cannot send “next-hop null0”... but:
- BGP can send “next-hop 192.0.2.1”
- And on each border router:
 - ip route 192.0.2.1 null0
- Router receives iBGP routing update:
 - “Route x.x.x.x next-hop 192.0.2.1” (comm: local-AS)
 - And it has an ip route 192.0.2.1 null0
 - Thus: x.x.x.x → null0



Discard Interface

- JUNOS supports the use of a 'discard' interface
 - Can assign filters to get granular filters, counting and sampling logs too



Discard Interface Configuration

```
Show configuration interfaces dsc
```

```
Unit 0 {  
  family inet {  
    address 192.168.1.1/32 {  
      destination 192.168.1.2;  
    }  
    address 192.168.1.3/32 {  
      destination 192.168.1.4;  
    }  
  }  
}
```

```
Sh interface terse
```

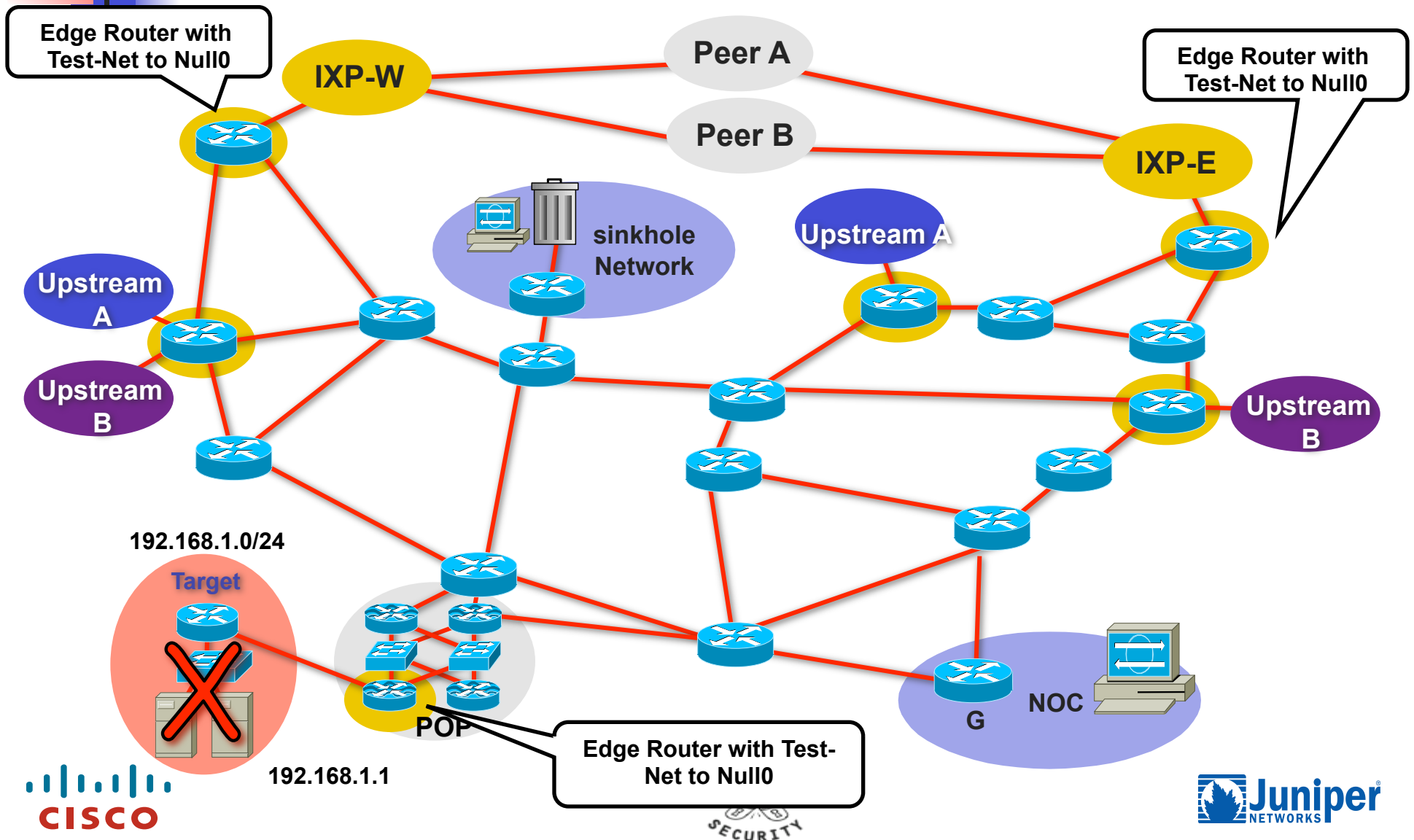
Interface	Admin	Link	Proto	Local	remote
Dsc.0		Up	Up	Inet 192.168.1.1	
192.168.1.2					

Step 1- Prepare all the Routers w/ Trigger

- Select a small block that will not be used for anything other than black hole filtering. Test Net (192.0.2.0/24) is optimal since it should not be on the Net and is not really used.
- Put a static route with Test Net – 192.0.2.0/24 to Null 0 on every edge router on the network.

```
ip route 192.0.2.1 255.255.255.255 Null0 255  
ip route 192.0.2.2 255.255.255.255 Null0 199  
ip route 192.0.2.3 255.255.255.255 Null0 50
```

Step 1- Prepare all the Routers w/ Trigger





Step 2 – Prepare the Trigger Router

- The trigger router is the device that will inject the iBGP announcement into the ISP's Network.
- iBGP neighbor - it can (and probably should) be a route-reflector client, no need for full mesh
- Can be a separate router or Arbor's Peakflow DoS anomaly-detection system - if a separate router, something small like a 2600 will do (it doesn't need to receive BGP routes, only send them; use a prefix-list to ensure that it doesn't end up redistributing routes)
- Can be a production router (some ISPs do this, but not the recommended approach)
- Can be a workstation with Zebra/Quagga (interface with Perl scripts and other tools).

Trigger Router's Config

Redistribute
Static with a
route-map

```
router bgp 109
```

```
redistribute static route-map static-to-bgp
```

```
.
```

```
!
```

```
route-map static-to-bgp permit 10
```

Match
Static
Route
Tag

```
match tag 66
```

```
set ip next-hop 192.0.2.1
```

```
set local-preference 50
```

```
set community no-export
```

```
set origin igp
```

```
!
```

```
Route-map static-to-bgp permit 20
```

Set Next-Hop
to the Trigger



Step 3 – Activate the Black Hole

- ISP adds a static route of the destination address they wish to black hole to the advertising router. The static is added with the “tag 66” to keep it separate from other statics on the router.
 - `ip route 192.168.1.1 255.255.255.255 Null0 Tag 66`
 - BGP Advertisement goes out to all BGP-speaking routers which peer with the trigger.
- Routers hear the announcement, glue it to the existing static on the route, and changes the next-hop for the BGP advertised route to Null0 – triggering black hole routing.



Activate the Black Hole

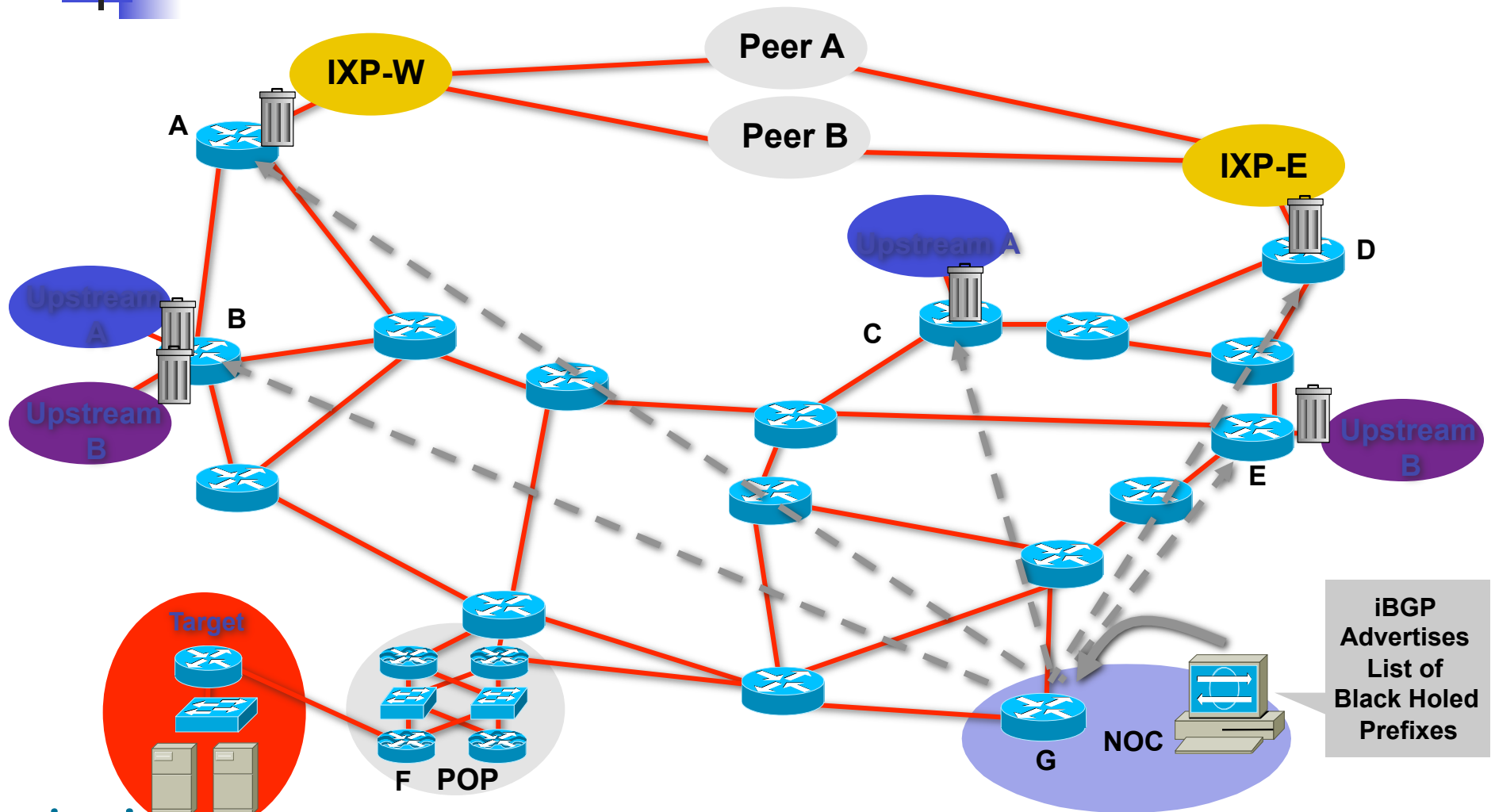
BGP Sent – 192.168.1.1 Next-Hop = 192.0.2.1

Static Route in Edge Router – 192.0.2.1 = Null0

192.168.1.1 = 192.0.2.1 = Null0

Next hop of 192.168.1.1 is now equal to Null0

Activate the Black Hole



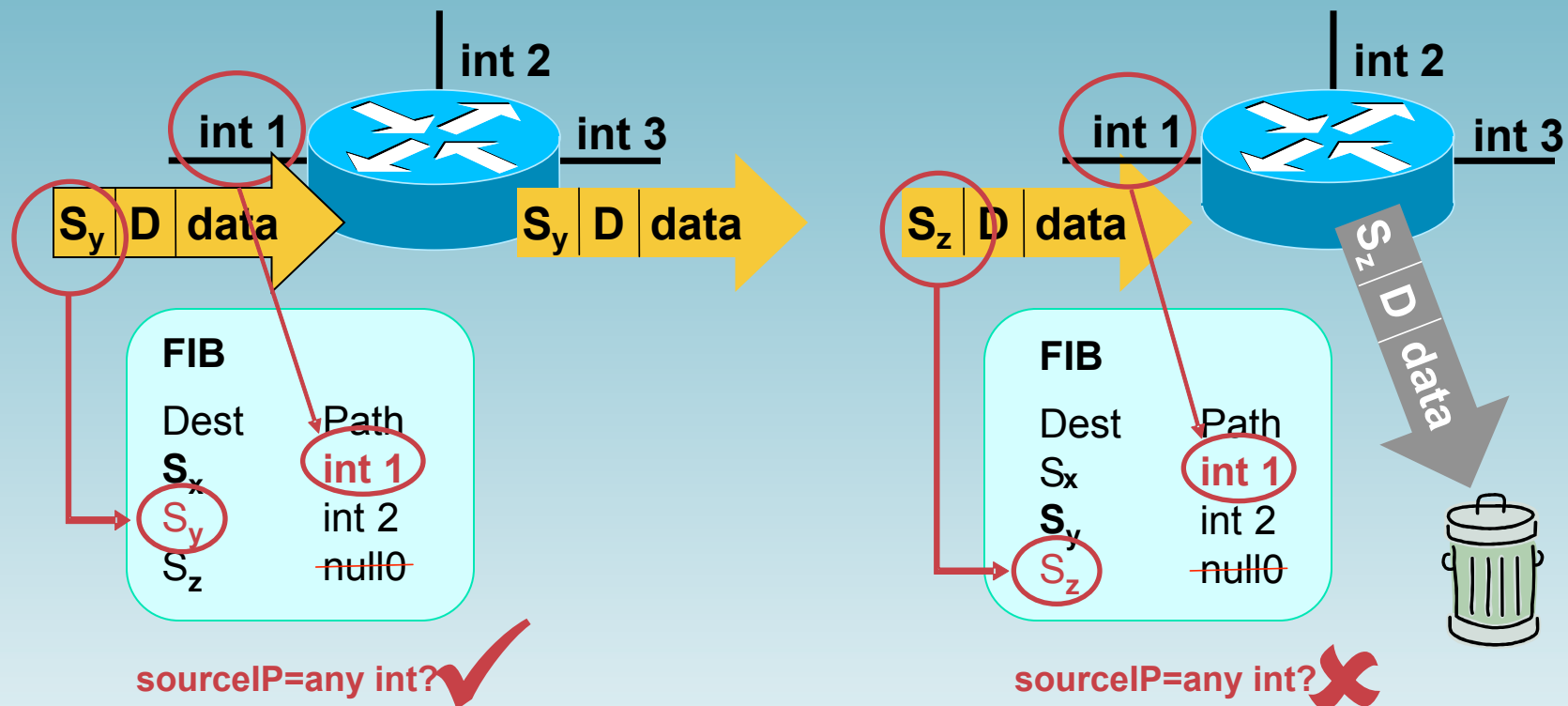


Flipping it Around: Triggered Source Drops

- Dropping on destination is very important
 - Dropping on source is often what we really need
- Reacting using source address provides some interesting options:
 - Stop the attack without blackholing real services
 - Filter command and control servers
 - Filter (contain) infected end stations
- Must be rapid and scaleable
 - Leverage pervasive BGP again

uRPF – Loose Mode

router(config-if)# ip verify unicast source reachable-via any



IP verify unicast source reachable – via any



Source Based RTBH Filtering

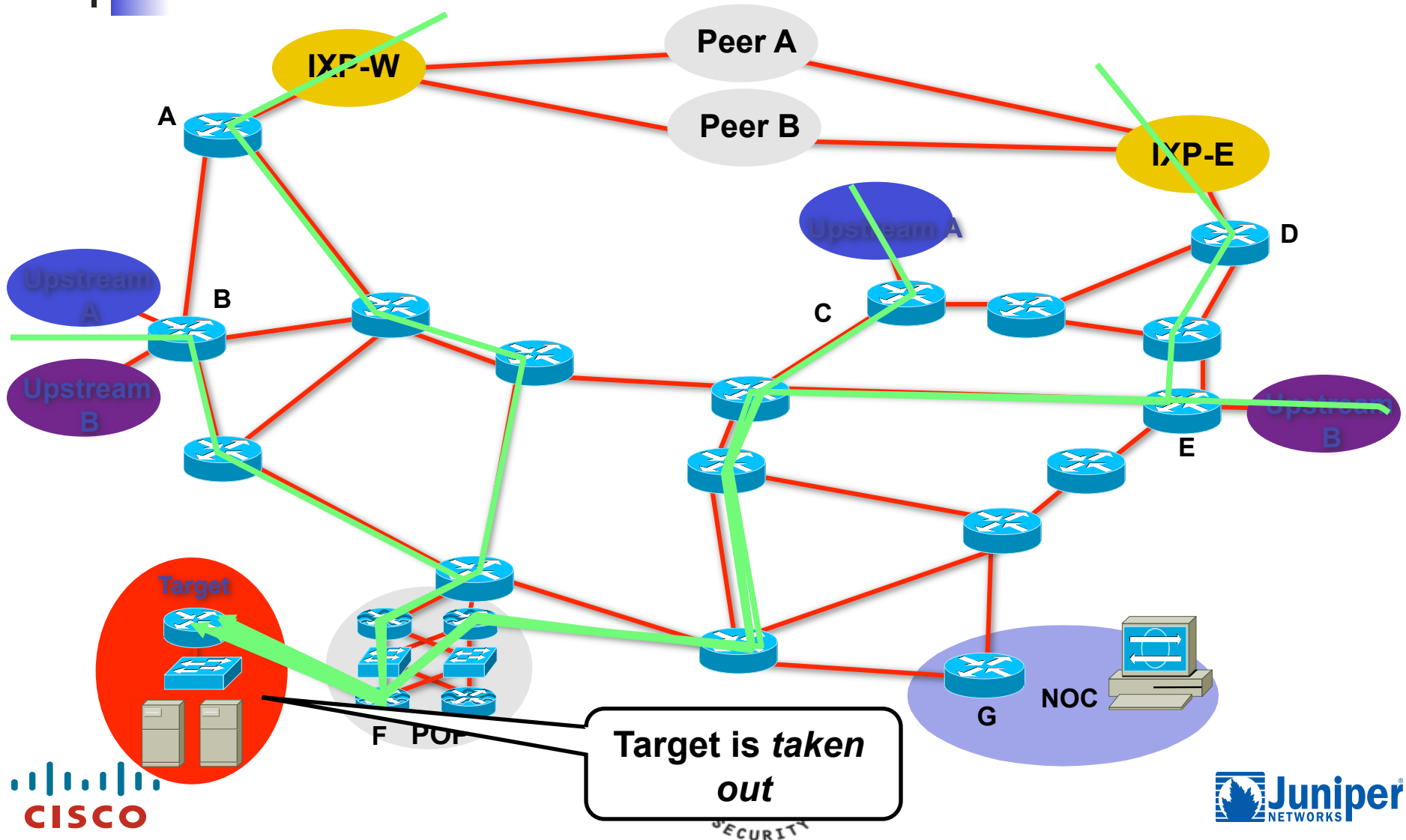
- Uses the same architecture as destination based filtering + unicast RPF
 - Edge routers must have static in place
 - They also require unicast RPF
 - BGP trigger sets next hop -- in this case the "victim" is the source we want to drop



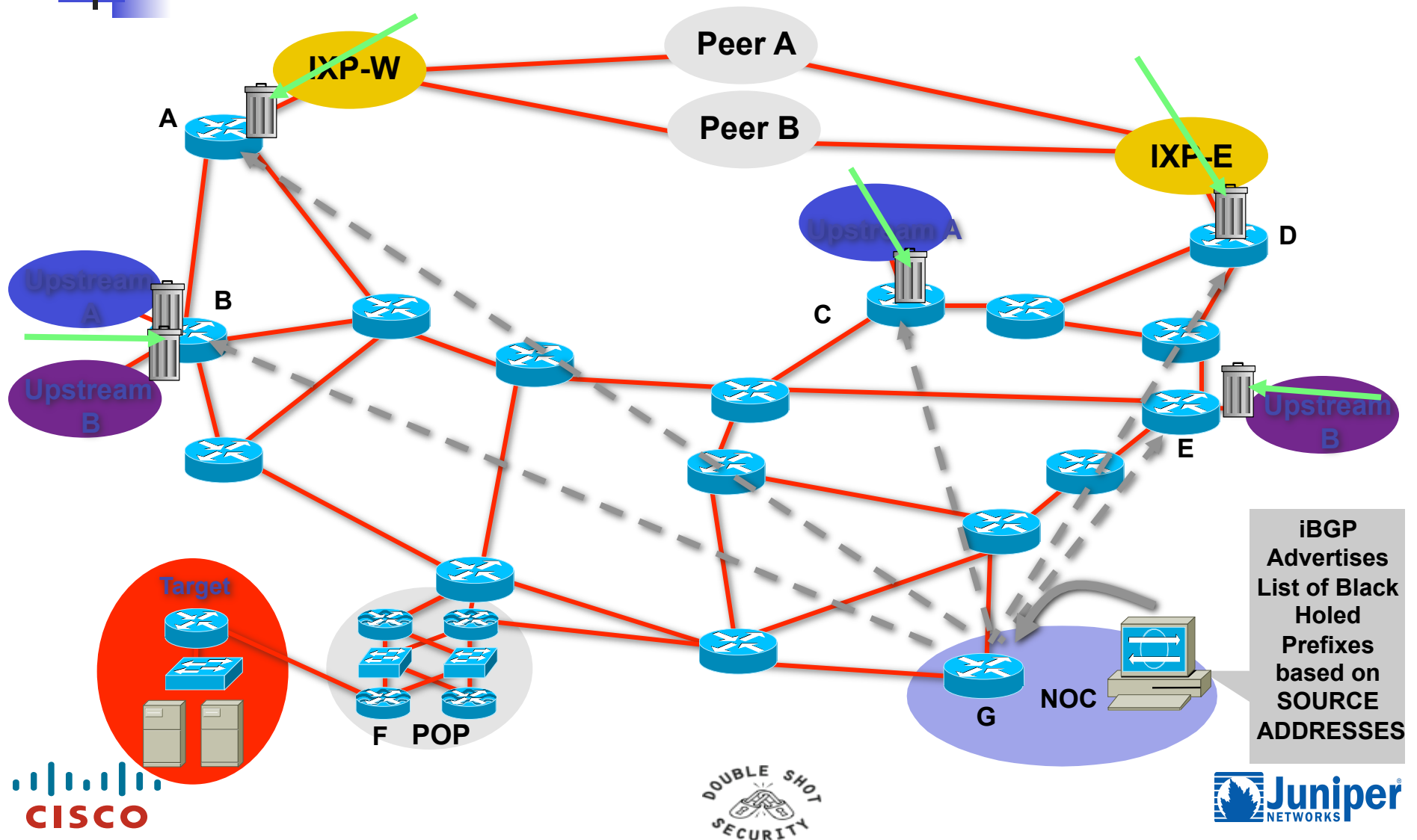
Source Based RTBH Filtering

- 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!

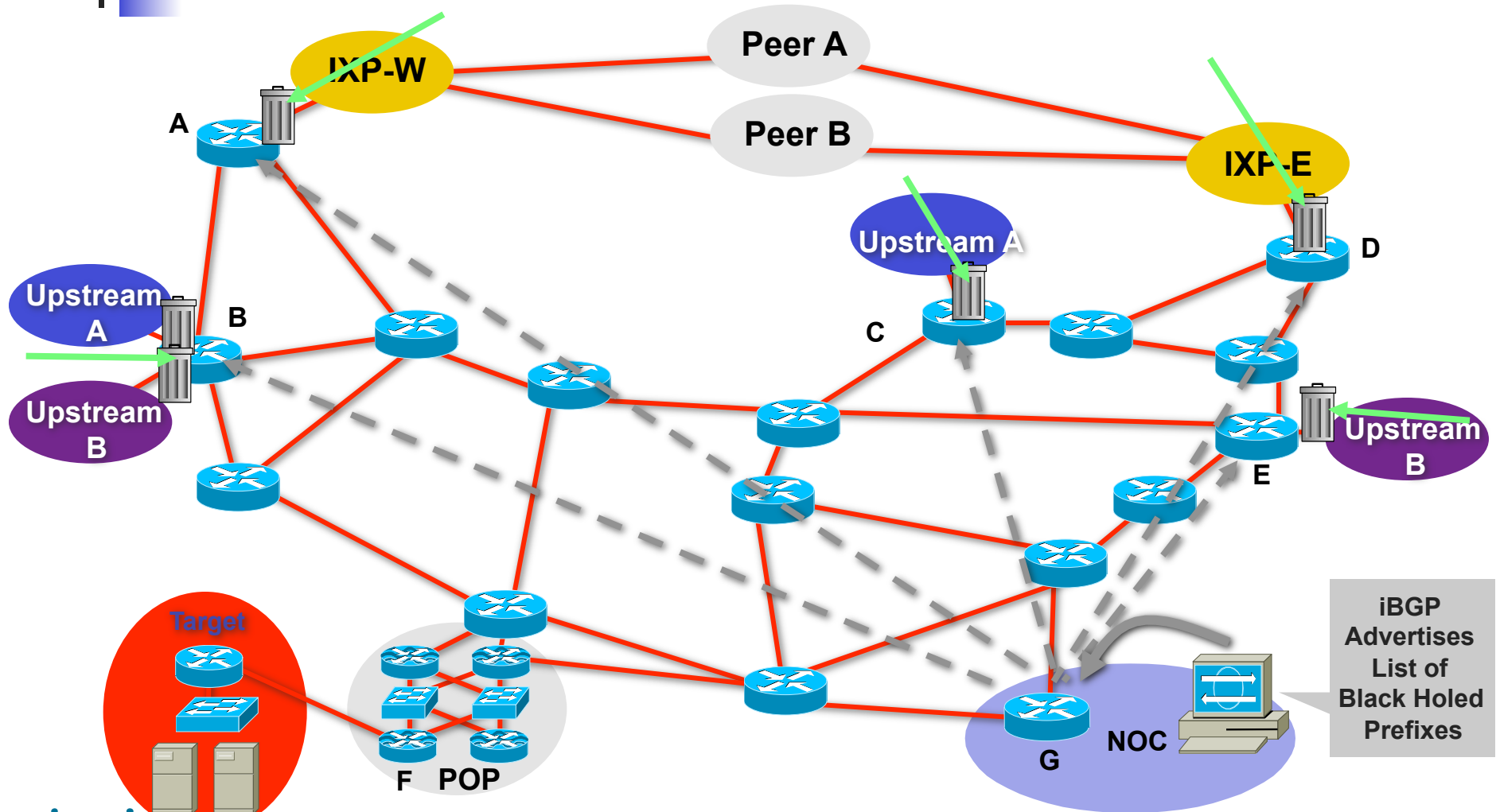
Customer is DOSed - Before



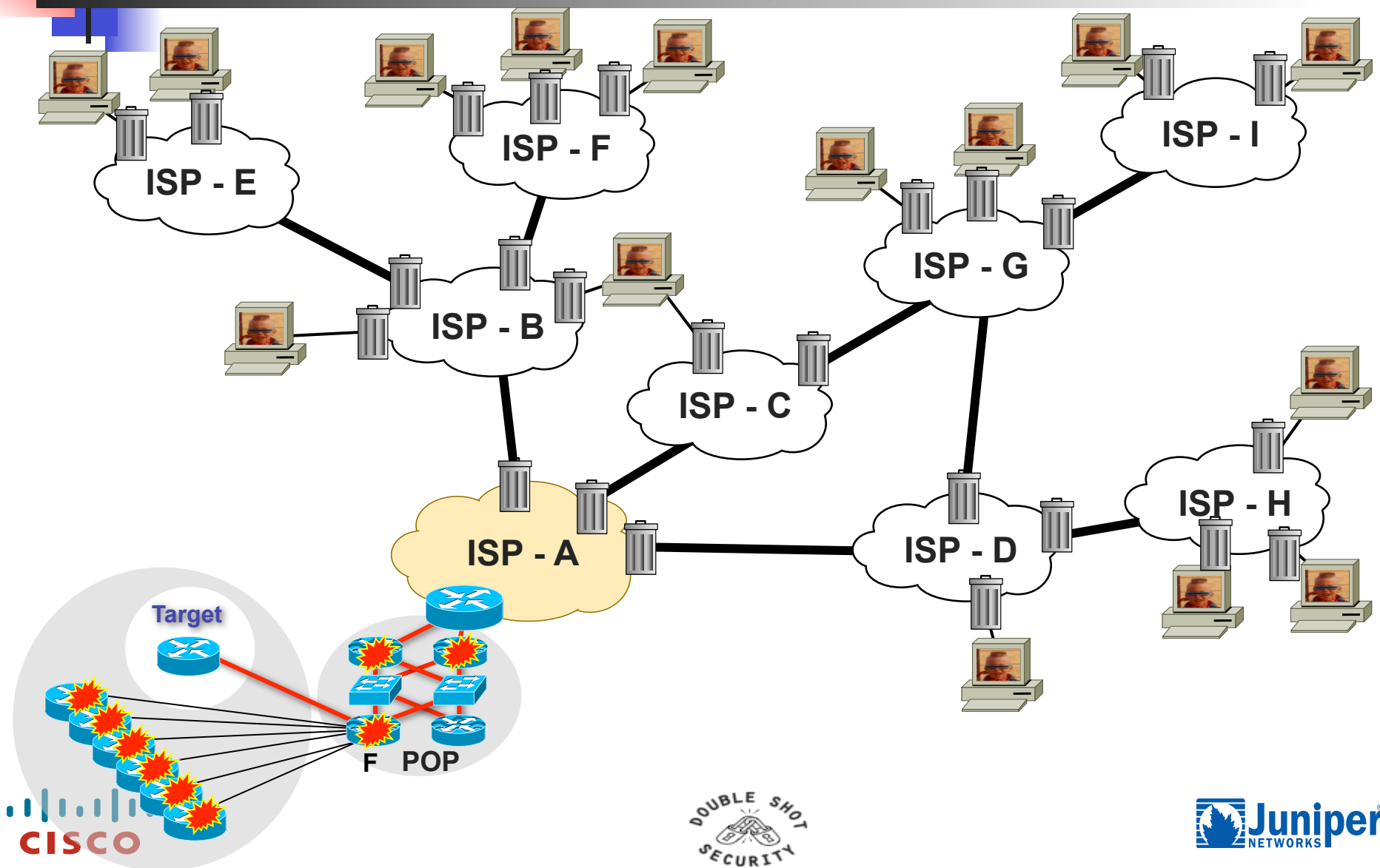
Customer is DOSed – After



Customer is DOSed – After – Packet Drops Pushed to the Edge



Inter-Provider Mitigation

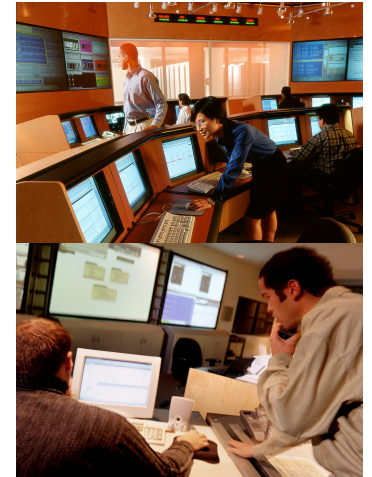
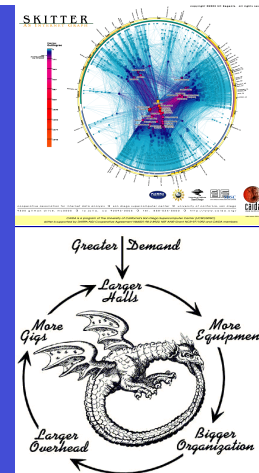




What can you do to help?

- Remote Triggered Black Hole Filtering is the most common ISP DOS/DDOS mitigation tool.
- Prepare your network:
 - <ftp://ftp-eng.cisco.com/cons/isp/essentials/> (has whitepaper)
 - <ftp://ftp-eng.cisco.com/cons/isp/security/> (has PDF Presentations)
 - NANOG Tutorial:
 - <http://www.nanog.org/mtg-0110/greene.html> (has public VOD with UUNET)

Sink Holes

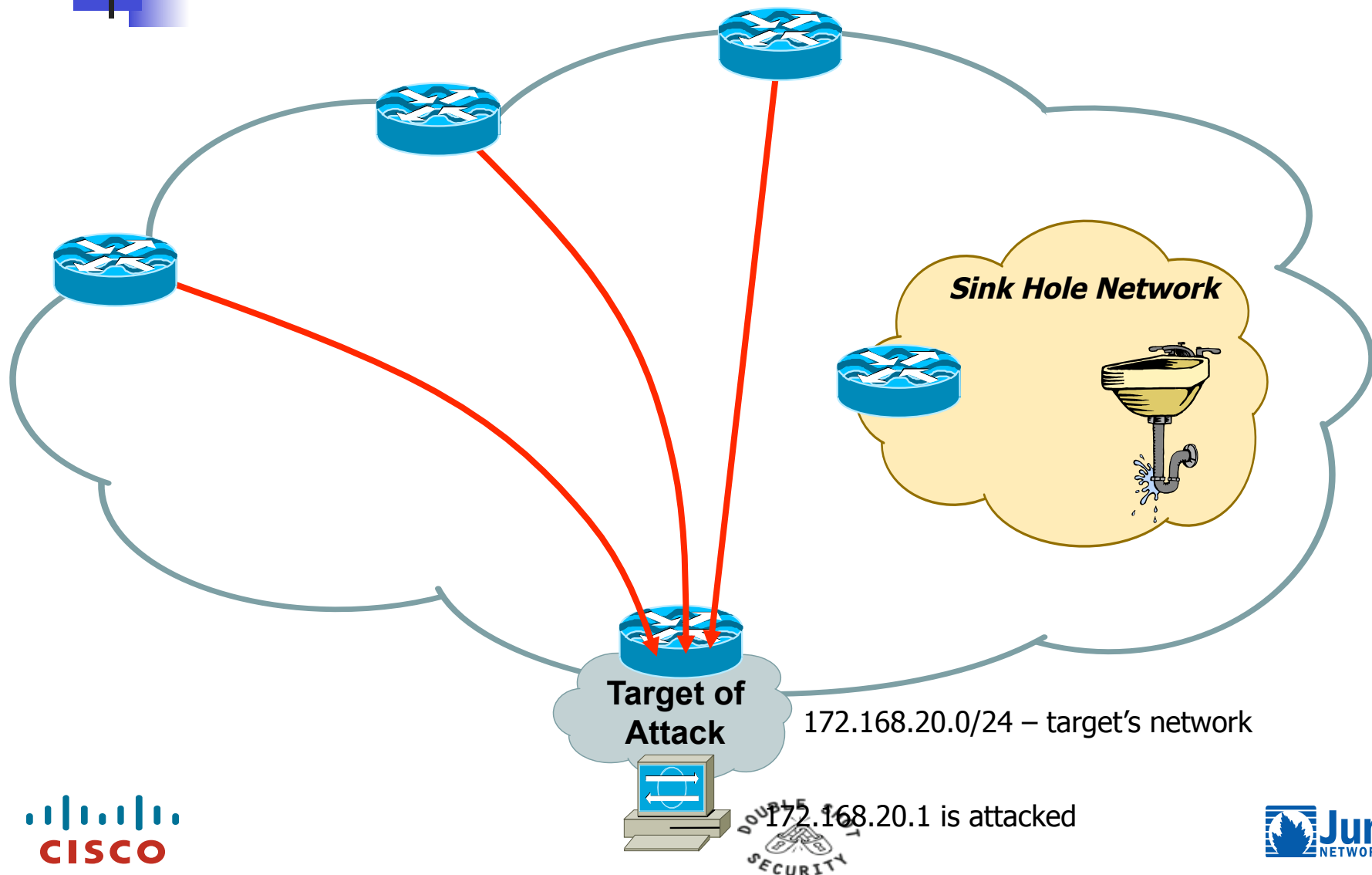




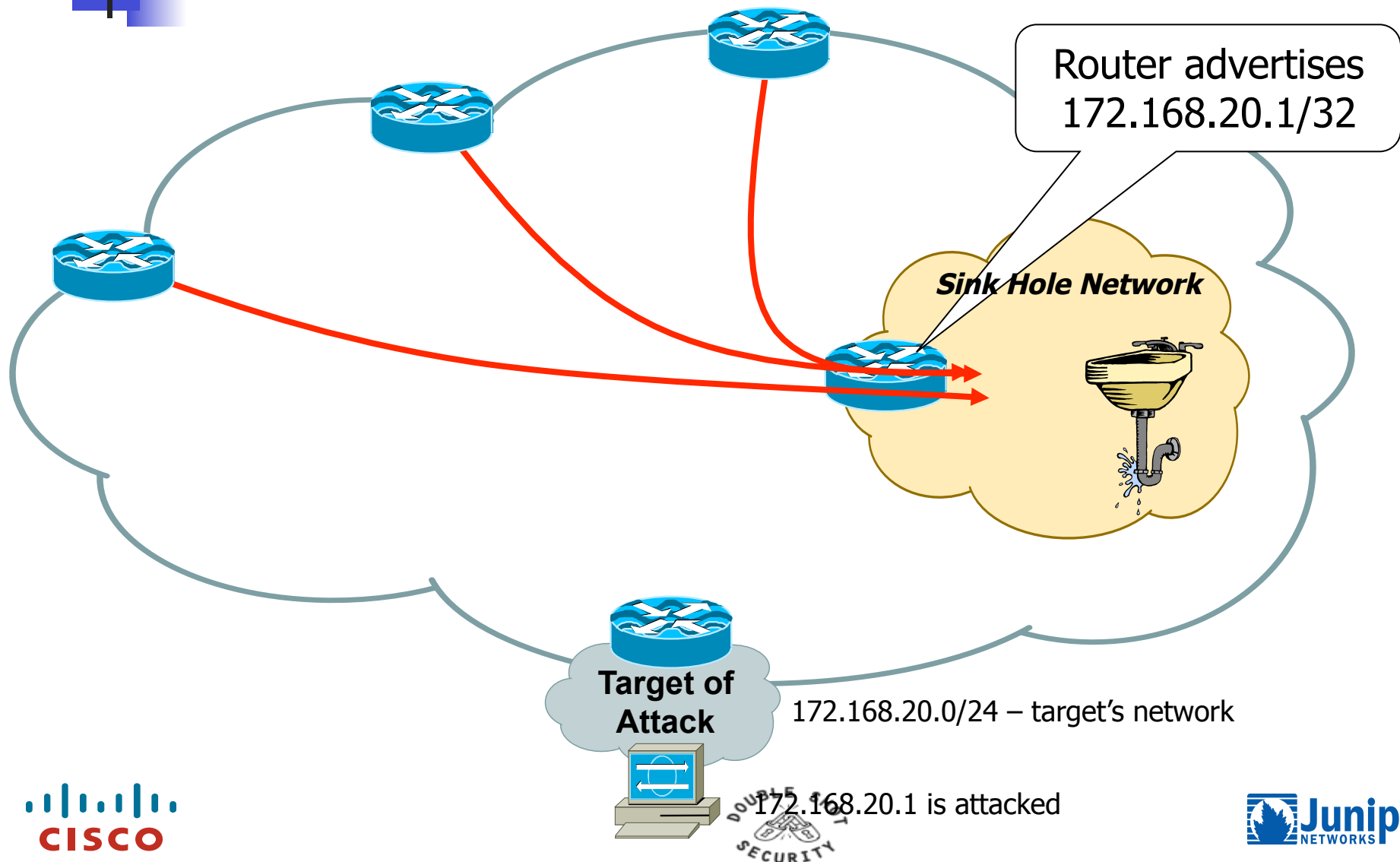
Sink Hole Routers/Networks

- Sink Holes are a *Swiss Army Knife* security tool.
 - BGP speaking Router or Workstation that built to *suck in* attacks.
 - Used to redirect attacks away from the customer – working the attack on a router built to withstand the attack.
 - Used to monitor *attack noise, scans*, and other activity (via the advertisement of default)
 - <http://www.nanog.org/mtg-0306/sink.html>

Sink Hole Routers/Networks

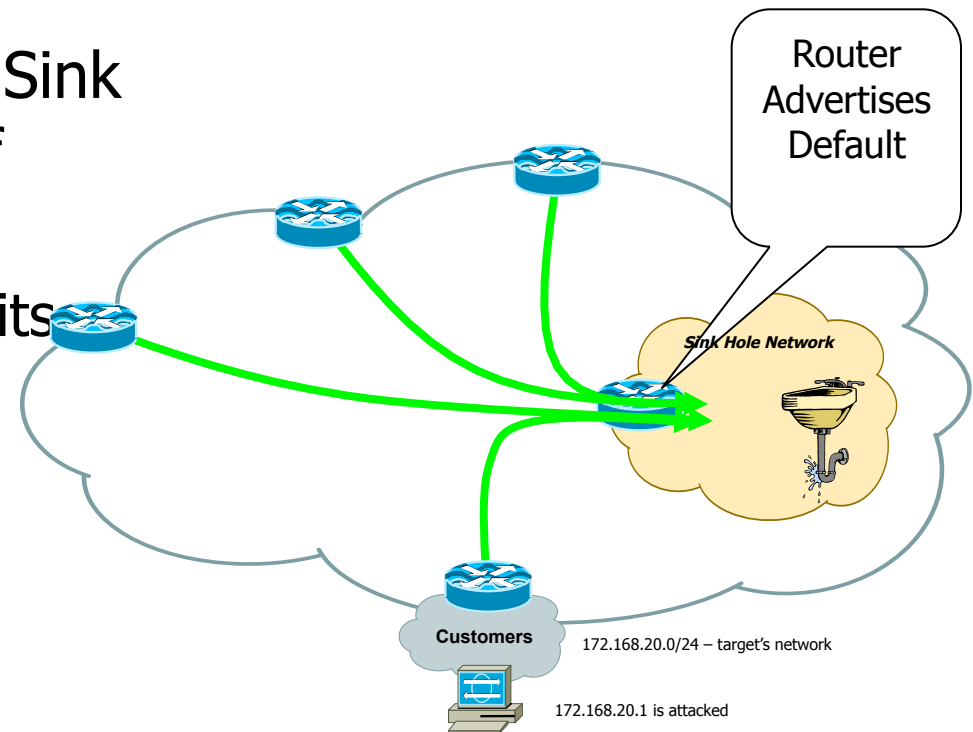


Sink Hole Routers/Networks

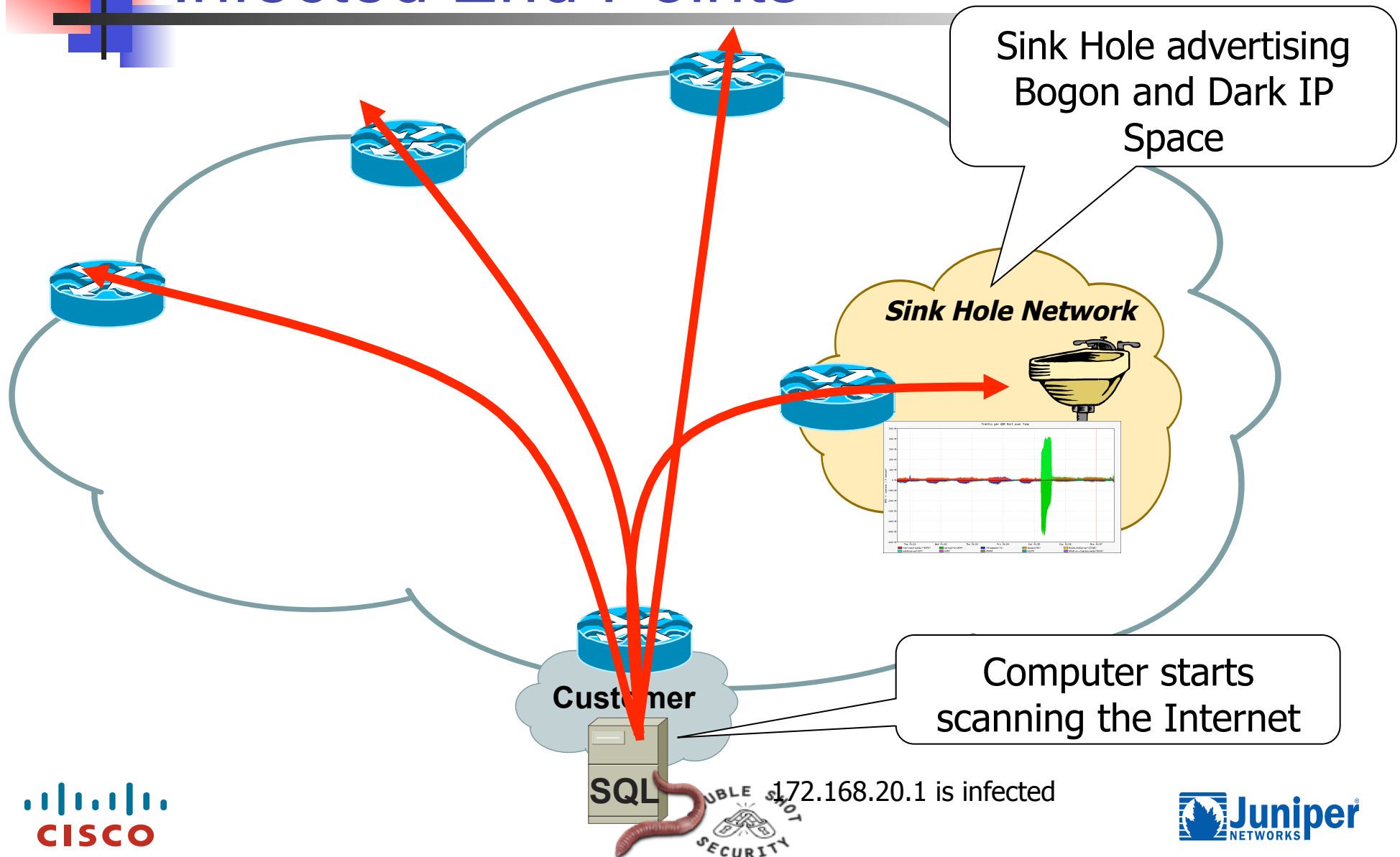


Sink Hole Routers/Networks

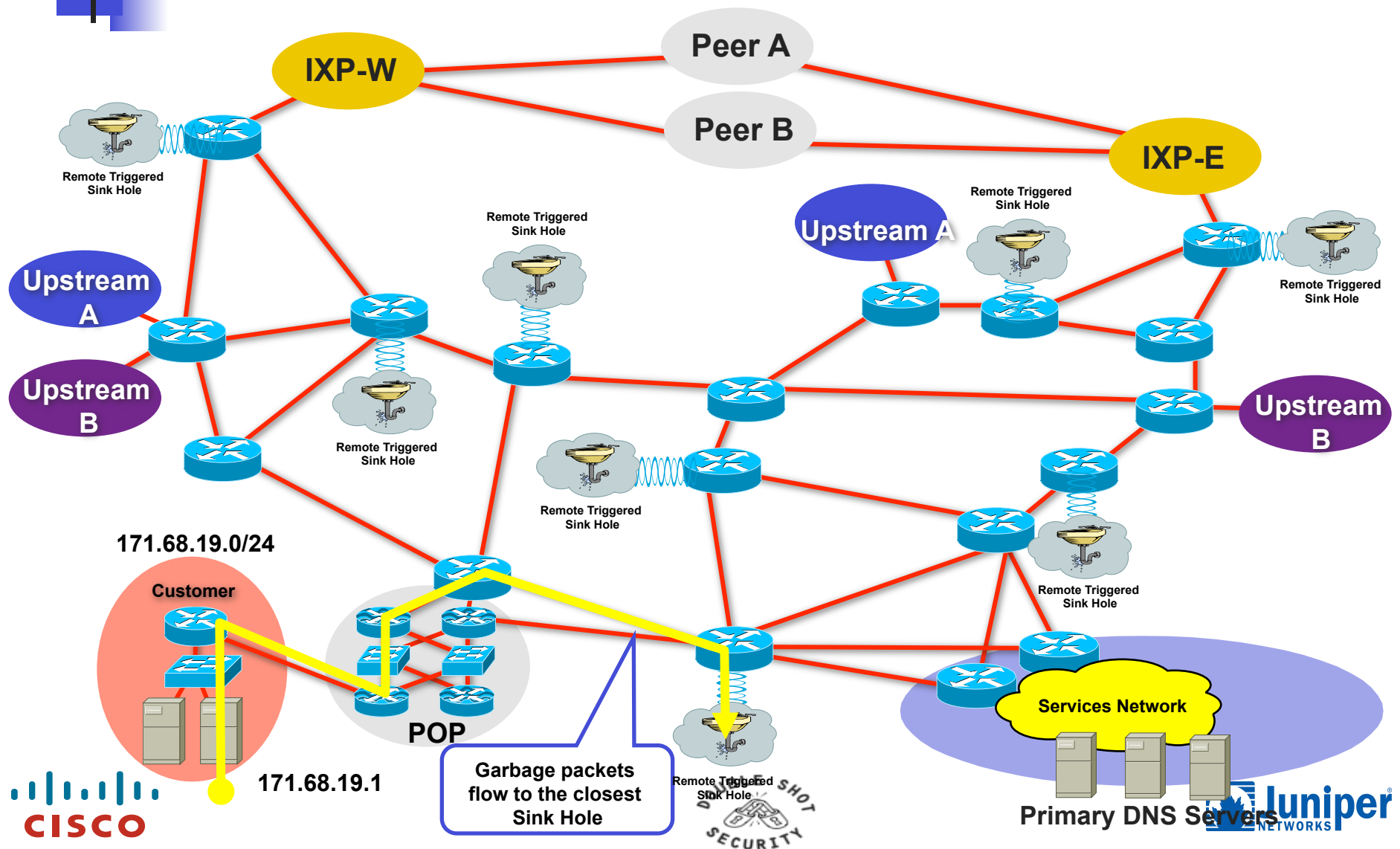
- Advertising Default from the Sink Hole will pull down all sort of *junk* traffic.
 - Customer Traffic when circuits flap.
 - Network Scans
 - Failed Attacks
 - Code Red/NIMDA
 - Backscatter
- Can place tracking tools and IDA in the Sink Hole network to monitor the noise.



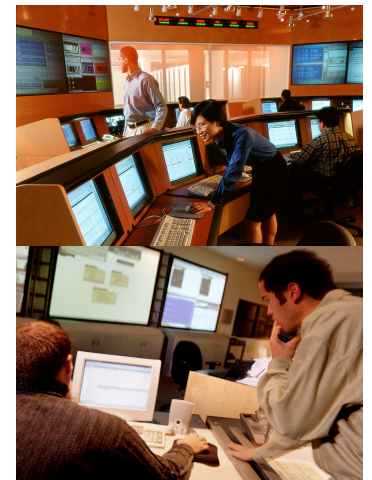
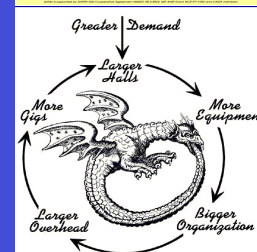
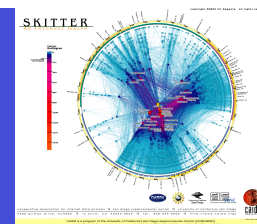
Infected End Points



Anycast Sink Holes



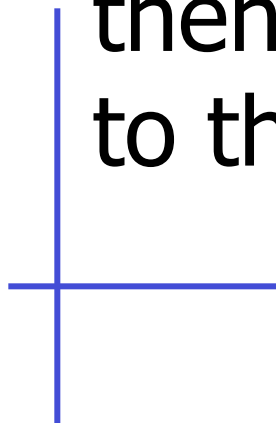
Source Address Validation





BCP 38 Ingress Packet Filtering

Your customers should not be sending any IP packets out to the Internet with a source address other than the address you have allocated to them!





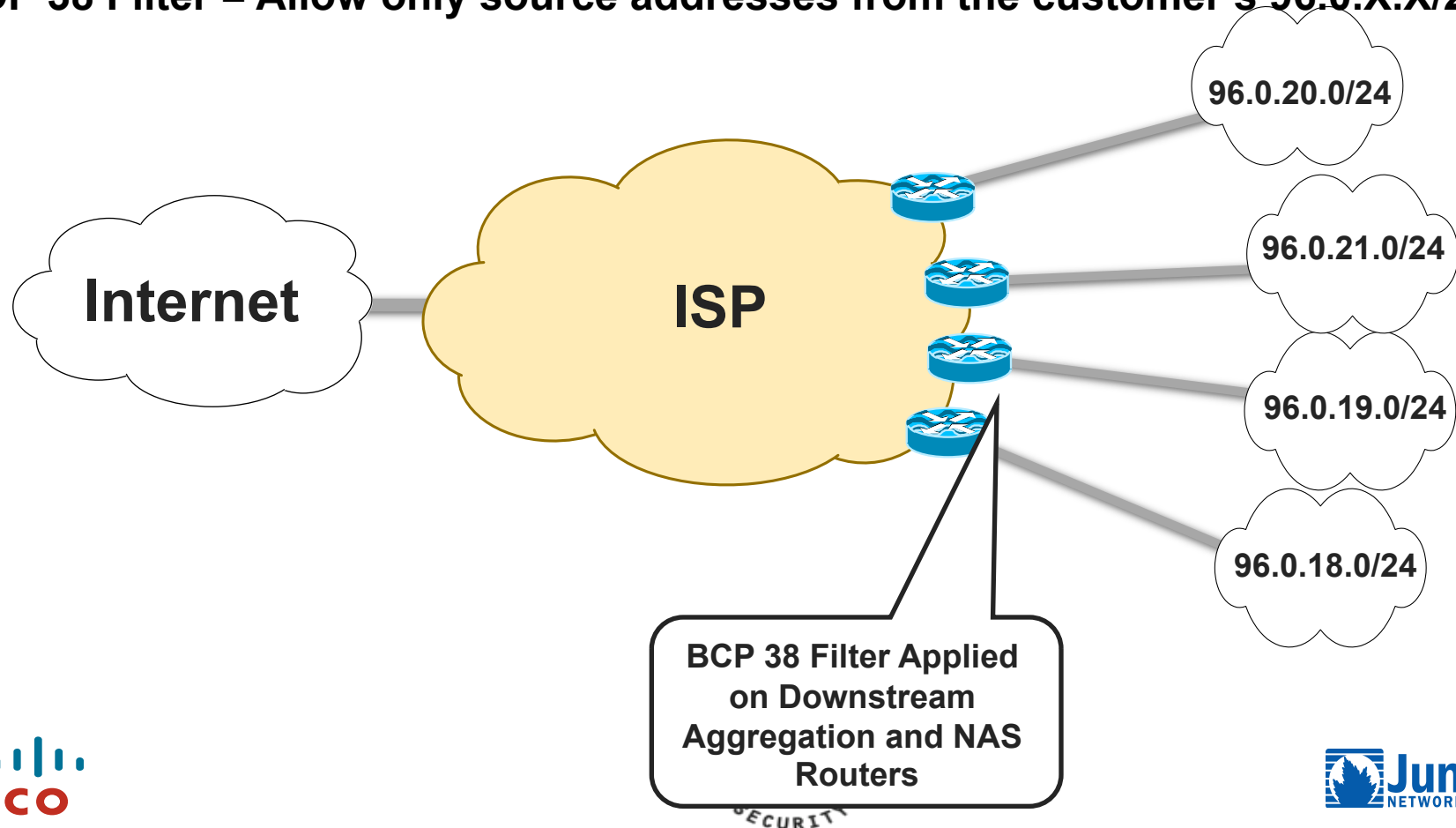
BCP 38 Ingress Packet Filtering

- BCP 38/ RFC 2827
- Title: Network Ingress Filtering: Defeating Denial of Service Attacks which Employ IP Source Address Spoofing
- Author(s): P. Ferguson, D. Senie

BCP 38 Ingress Packet Filtering

ISP's Customer Allocation Block: 96.0.0.0/19

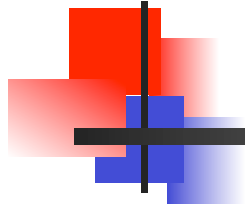
BCP 38 Filter = Allow only source addresses from the customer's 96.0.X.X/24





BCP 38 Packet Filtering: Principles

- Filter as close to the edge as possible
- Filter as precisely as possible
- Filter both source and destination where possible



Many *Working* Techniques

- Static access list on the edge of the network
- Dynamic access list with AAA profiles
- Unicast RPF
- Cable Source Verify (MAC & IP)
- Packet Cable Multimedia (PCMM)
- IP Source Verify (MAC & IP)



Source Address Validation Works

- Successful ISPs have extremely conservative engineering practices.
- Operational Confidence in the equipment, functionality, and features are a prerequisite to any new configs on a router.
- The core reason why ISPs have not been turning on Source Address Validation is their lack of *Operational Confidence*.



One Major ISP's Example - uRPF

- Month 1 – Cisco Lab Test and Education to help the customer gain confidence in uRPF.
- Month 2 – One port on one router – turning uRPF Strict Mode on a 16xOC3 Engine 2 LC (Cisco 12000)
- Month 3 – One LC on one router – 16xOC3.
- Month 4 – One router all customer facing LCs
- Month 5 – One POP – all customer facing LCs
- Month 6 – Several routers through out the network (other POPs)
- Month 7 – Adopted as standard config for all new customer circuits. Will migrate older customer over time.



One Major ISP's Example - uRPF

- Lessons Learned:
 - It took time and patience.
 - uRPF did not work for all customers. That is OK, uRPF is not suppose to be a *universal solution*.
 - Going slow and steady allowed the operations team to *gain a feel* of the feature's performance envelope --- with out putting the network at risk.
- It works! A year later it is a standard config with over 40K ports running uRPF Strict or Loose Mode.



What can you do to help?

- Cut the excuses! BCP 38 is an operational reality!
- Walk them through source address validation techniques, see which ones will work for you, and do not expect more than a 80% success rate.
- Find ways to gain operational confidence in the BCP 38 techniques.
- Source Address validation works – it just take patience and persistence.