

CISCO SYSTEMS



Essential IOS Features Every ISP Should Consider

*Lessons from people who have been operating
backbones since the early days of the Net*

Version 2.82

Saturday, February 19, 2000

TABLE OF CONTENTS

TABLE OF CONTENTS	3
LIST OF FIGURES	8
ACKNOWLEDGEMENTS	9
INTRODUCTION	10
SOFTWARE AND ROUTER MANAGEMENT	11
WHICH IOS VERSION SHOULD I BE USING?	11
<i>Where to Get Information on 11.1CC?</i>	12
<i>Where to Get Information on 12.0S</i>	12
<i>Further Reference on IOS Software Releases</i>	12
IOS SOFTWARE MANAGEMENT.....	13
<i>Flash Memory</i>	13
<i>System Memory</i>	14
<i>When and How to Upgrade?</i>	15
<i>Copying new images to FLASH Memory</i>	15
CONFIGURATION MANAGEMENT	17
<i>NVRAM and TFTPserver</i>	17
<i>Large Configurations</i>	17
DETAILED LOGGING	18
<i>Analysing Syslog Data</i>	19
NETWORK TIME PROTOCOL (NTP)	19
<i>NTP Architecture</i>	19
<i>Client/Server Models and Association Modes</i>	20
<i>Implementing NTP on an ISP's Routers</i>	21
<i>NTP Deployment Examples</i>	21
<i>Further NTP References</i>	22
SIMPLE NETWORK MANAGEMENT PROTOCOL (SNMP)	22
HTTP SERVER	23
CORE DUMPS.....	24
GENERAL FEATURES	25
COMMAND LINE INTERFACE	25
<i>Editing Keys</i>	25
<i>CLI String Search</i>	25
INTERFACE CONFIGURATION.....	26
<i>Description</i>	26
<i>Bandwidth</i>	26
<i>IP Unnumbered</i>	27
Caveats	27
<i>A Full Example</i>	27
NETFLOW.....	28
DNS AND ROUTERS	30
IOS AND LOOPBACK INTERFACES	31
<i>Background</i>	31
<i>BGP Update-Source</i>	32
<i>Router ID</i>	32
<i>IP Unnumbered Interfaces</i>	32

<i>Exception Dumps by FTP</i>	33
<i>TFTP-SERVER Access</i>	33
<i>SNMP-SERVER Access</i>	33
<i>TACACS/RADIUS-Server Source Interface</i>	33
<i>NetFlow Flow-Export</i>	34
<i>NTP Source Interface</i>	34
<i>SYSLOG Source Interface</i>	35
<i>Telnet to the Router</i>	35
<i>RCMD to the router</i>	36
SECURITY	37
SECURITY FOR AN ISP.....	37
GLOBAL SERVICES THAT ARE NOT NEEDED OR ARE A SECURITY RISK.....	38
INTERFACE SERVICES THAT ARE NOT NEEDED OR ARE A SECURITY RISK.....	38
CISCO DISCOVERY PROTOCOL.....	39
LOGIN BANNERS.....	39
USE ENABLE SECRET.....	40
TURN ON NAGLE.....	41
THE IDENT FEATURE.....	41
SYSTEM ACCESS.....	41
<i>Principles</i>	41
<i>VTY and Console Port Timeouts</i>	42
<i>Access List on the VTY Ports</i>	42
<i>VTY Access and SSH</i>	43
<i>User authentication</i>	43
<i>Using AAA to Secure the Router</i>	44
<i>Router Command Auditing</i>	45
<i>Full Example</i>	45
EGRESS AND INGRESS FILTERING.....	46
<i>Ingress and Egress Route Filtering</i>	46
<i>Ingress and Egress Packet Filtering</i>	47
Egress Filtering – Preventing Transmission of Invalid IP Addresses.....	47
Ingress Filtering – Preventing Reception of Invalid IP Addresses.....	48
<i>Unicast RPF – Reverse Path Forwarding</i>	49
RPF Configuration Details.....	51
RPF Implementation Notes.....	52
Unicast RPF and Routing Asymmetry.....	52
Routing Tables Requirements.....	54
Unicast RPF Exceptions.....	54
<i>Unicast RPF Example – Putting it all together</i>	54
<i>Other Considerations</i>	55
AUTHENTICATING ROUTING PROTOCOL UPDATES.....	55
<i>Benefits of Neighbour Authentication</i>	55
<i>Protocols That Use Neighbour Authentication</i>	55
<i>When to Configure Neighbour Authentication</i>	56
<i>How Neighbour Authentication Works</i>	56
<i>Plain Text Authentication</i>	56
<i>MD5 Authentication</i>	57
CAR AS A SMURF REACTION/PREVENTION TOOL.....	57
<i>What is a SMURF or FRAG Attack?</i>	57
<i>Passive SMURF Defences</i>	57
<i>Active SMURF Defences</i>	58
Rate Limiting with CAR.....	58
ROUTING	60
HOT STANDBY ROUTING PROTOCOL.....	60

CIDR FEATURES	61
SELECTIVE PACKET DISCARD.....	62
IP SOURCE ROUTING.....	63
CONFIGURING ROUTING PROTOCOLS	63
<i>Router ID</i>	63
<i>Choosing an IGP</i>	64
<i>Configuring an IGP</i>	64
<i>Putting Prefixes into the IGP</i>	64
The Network Statement	64
Redistribute Connected into an IGP	65
Redistribute Static into an IGP	65
Redistribute <anything> into an IGP	65
<i>IGP Summarisation</i>	65
<i>IGP Adjacency Change Logging</i>	65
<i>Putting Prefixes into BGP</i>	66
The Network Statement	66
Redistribute Connected into BGP.....	66
Redistribute Static into BGP.....	66
Redistribute <anything> into BGP	67
BGP FEATURES AND COMMANDS	67
<i>Stable iBGP Configuration</i>	68
<i>BGP Auto Summary</i>	68
<i>BGP Synchronisation</i>	69
<i>BGP Community Format</i>	69
<i>BGP Neighbour Shutdown</i>	69
<i>BGP Soft-Reconfiguration</i>	70
<i>Next-Hop-Self</i>	70
External connections	71
Aggregation routers	71
<i>BGP Dampening</i>	71
<i>BGP Neighbour Authentication</i>	73
<i>Removing Private ASes</i>	74
<i>BGP local-as</i>	74
Configuration	74
Motivation	75
<i>BGP Neighbour Changes</i>	75
<i>Limiting the Number of Prefixes from a Neighbour</i>	75
<i>BGP Fast External Fallover</i>	76
<i>BGP Peer-group</i>	76
Summary	76
Requirements.....	76
Historical Limitations	77
Typical Peer-group Usage	77
BGP Peer-Group Examples	77
<i>Using Prefix-list in Route Filtering</i>	78
Introduction	78
Configuration Commands.....	78
Command Attributes	78
Configuration Examples	79
How Does Match Work.....	80
Show and Clear Commands.....	80
Using Prefix-list with BGP.....	80
Using Prefix-list in Route-map.....	81
Using Prefix-list in Other Routing Protocols.....	81
<i>BGP Conditional Advertisement</i>	82
Example.....	82
<i>BGP Route Refresh</i>	84

<i>BGP Outbound Route Filter Capability</i>	84
Configuration	84
Pushing out a Prefix-list ORF.....	85
Displaying Prefix-list ORF.....	85
<i>BGP Policy Accounting</i>	85
Overview	85
Configuration	86
Displaying BGP Policy Accounting status	86
Displaying BGP Policy Accounting statistics.....	87
FURTHER STUDY AND TECHNICAL REFERENCES	88
APPENDIX 1 – ACCESS LIST AND REGULAR EXPRESSIONS	89
ACCESS LIST TYPES	89
BASIC REGULAR EXPRESSIONS	90
APPENDIX 2 – CUT AND PASTE TEMPLATES	91
GENERAL SYSTEM TEMPLATE.....	91
GENERAL INTERFACE TEMPLATE.....	91
GENERAL SECURITY TEMPLATE.....	91
GENERAL BGP TEMPLATE	91
MARTIAN AND RFC1918 NETWORKS TEMPLATE	92
<i>IP Access-List Example</i>	92
<i>IP Prefix-List Example</i>	92
BGP FLAP DAMPENING CONFIGURATION	92
<i>IP Access-List Example</i>	92
<i>IP Prefix-List Example</i>	93
APPENDIX 3 – TRAFFIC ENGINEERING TOOLS	95
INTERNET TRAFFIC AND NETWORK ENGINEERING TOOLS.....	95
CAIDA.....	95
NetScarf/Sicon.....	95
NeTraMet/NetFlowMet.....	95
Cflowd.....	95
MRTG.....	96
RRDTool.....	96
Vulture.....	96
CMU SNMP.....	96
UCD SNMP (the successor to CMU SNMP).....	97
Gnuplot.....	97
NETSYS	97
SysMon.....	98
Treno	98
Scotty – Tcl Extensions for Network Management Applications.....	98
OTHER USEFUL TOOLS TO MANAGE YOUR NETWORK	98
RTRMon – A Tool for Router Monitoring and Manipulation.....	98
Cisco’s MIBs	99
SECURE SYSLOG (ssyslog).....	99
OVERALL INTERNET STATUS AND PERFORMANCE TOOLS.....	99
NetStat.....	99
WHAT OTHER ISPS ARE DOING.....	99
APPENDIX 4 – EXAMPLE ISP ACCESS SECURITY MIGRATION PLAN	103
PHASE ONE – CLOSE OFF ACCESS TO EVERYONE OUTSIDE YOUR CIDR BLOCK.....	103
PHASE TWO – ADD ANTI-SPOOFING FILTERS TO YOUR UPSTREAM GATEWAYS AND PEERING POINTS.....	104

Where to place the anti-spoofing packet filters? 105
PHASE THREE – CLOSE OFF NETWORK EQUIPMENT ACCESS TO EVERYONE EXCEPT THE NOC AND OTHER AUTHORISED
STAFF 106

LIST OF FIGURES

Figure 1 – IOS Roadmap.....	13
Figure 2 – Ingress and Egress Filtering	46
Figure 3 – Egress Filtering	48
Figure 4 – Ingress Filtering	49
Figure 5 – Unicast RPF validating IP source addresses	50
Figure 6 – Unicast RPF dropping packets which fail verification.....	50
Figure 7 – Unicast RPF Drop Counter	52
Figure 8 – Unicast RPF applied to Lease Line Customer Connections.....	53
Figure 9 – Unicast RPF applied to PSTN/ISDN Customer Connections	53
Figure 10 – How asymmetrical routing would not work with Unicast RPF.....	54
Figure 11 – How SMURF uses amplifiers	58
Figure 12 – Dual gateway LAN	61
Figure 13 – BGP Route Flap Dampening.....	73
Figure 14 – Conditional Advertisement – Steady State.....	82
Figure 15 – Conditional Advertisement – Failure Mode.....	83
Figure 16 – ISP Network Example.....	104
Figure 17 – Applying Anti-Spoofing Filters	105
Figure 18 – Closing off access to everyone except the NOC Staff.....	107

ACKNOWLEDGEMENTS

We would like to thank the following people for helping make suggestions, contributions, corrections, and their deep real world operational experience with the Internet. Their willingness to help others *do the right thing* is one of the reasons for the Internet's success.

Bruce R. Babcock [bbabcock@cisco.com]
Tony Barber [tonyb@uk.uu.net]
Enke Chen [echen@cisco.com]
Paul Ferguson [ferguson@cisco.com]
Dorian R. Kim [dorian@blackrose.org]
Andrew Partan [asp@partan.com]

Any comments, questions, updates, or corrections should be sent to:

Barry Raveendran Greene bgreene@cisco.com
Philip Smith pfs@cisco.com

INTRODUCTION

Cisco Systems has a tremendous range of features built into the IOS. The extensive feature set is excellent for Network Engineers, giving them a large number of options and capabilities which can be designed into their network. At the same time, the huge feature set can be a problem. Network Engineers have a very difficult time keeping up with all the new IOS features. Many do not know how, when, and where to deploy the various features in their network. Experienced Network Engineers building the Internet are not exempt. This document has been written to highlight many of the key IOS features in everyday use in the major ISP backbones of the world. Judicious study and implementation of these *IOS pearls* will help to prevent problems, increase security, improve performance, and ensure the operational stability of the Internet.

NOTE: This document and its recommendations focus on Internet Service Providers – not the general Internet population. This point of view needs to be understood by the person using the techniques described in this whitepaper for their network.

This paper has four sections as well as several appendixes that give the reader further information, tips, and templates relating to what has been covered in the paper. These sections are:

- Software and Router Management
- General Features
- Security
- Routing

If you have questions on any of the materials in this whitepaper, please refer to the following:

- Cisco System's Documentation. (available free via <http://www.cisco.com/univercd/>)
- Cisco Connection On-line. (<http://www.cisco.com>)
- Local Cisco Systems' support channels,
- Public discussion lists. The list that specifically focuses on ISPs who use Cisco Systems equipment is *Cisco NSP* – hosted by Jared Mauch at Nether.Net¹.

An up to date copy of this document can be found at <http://www.cisco.com/public/cons/isp>. This page also has related material referenced in this document, plus other information which may be useful for Internet Service Providers.

¹ CISCO NSP is a mailing list which has been created specifically by Internet Service Providers to discuss Cisco Systems products and their use. To subscribe, send an e-mail to: majordomo@puck.nether.net with a message body containing: *subscribe cisco-nsp*

SOFTWARE AND ROUTER MANAGEMENT

Which IOS version should I be using?

ISPs and NSPs operate in an environment of constant change, exponential growth, and unpredictable threats to the stability of their backbone. The last thing an Internet backbone engineer needs is buggy or unstable code on their routers. As in any commercial grade service providing infrastructure, the equipment forming that infrastructure requires stable operating software. Stable software needs to have rapid updates with fixes to bugs that have been identified. This stable code needs to have the latest features – critical to their operations – added long before the rest of Cisco’s enterprise customers see them. (Herein is a key difference between enterprise and Internet service provision. The former demands stability and change only when necessitated, while the latter demands stability, yet market leadership when it comes to new features.) ISPs need to access this stable code via the Internet with out the traditional hassles of obtaining a software upgrade. Bottom line is that ISPs require an IOS code train specific to their needs.

This is exactly what has happened. Mid way through the life of the 10.3 software train, Cisco created a branch of IOS that catered specifically to ISPs’ requirements. The key characteristics were IP-centric code base, stability, quick bug fixes, easy access to the key development engineers, and rapid feature additions. The so-called “isp-geeks” software started life as an unofficial ISP software issue, but with the arrival of the 11.1 software train, has matured and developed into a release system specifically targeted at Internet Service Providers. As IOS becomes more and more feature rich, this ISP software train has been further developed and enhanced and now provides a very well developed and stable platform for all Internet Service Providers.

Along with the development of specific IOS images for ISPs, the Service Provider image was added to all Cisco IOS software released. This software is based on the IP-only image but with additional features for Service Providers. Such software can be recognised by the “-p-” in the image name. This image is usually all that any ISP needs to run. These images cannot be ordered at time of router purchase but can be downloaded from CCO prior to deployment of the router in service. For example, a 7200 which an ISP orders may come with the **c7200-i-mz.120-6** image – this image should be replaced with the Service Provider equivalent, **c7200-p-mz.120-6**. These Service Provider “-p-” images are built for all supported router platforms unlike the more limited platform support available on the ISP release trains.

At the time of writing, the recommended IOS branches for ISPs are:

- ✓ **11.1CA** – Old recommended release for ISPs with 7500s, 7200s, and RP/RSP7000s.
- ✓ **11.1CC** – Current recommended release for ISPs with 7500s, 7200s, and RSP7000s.
- ✓ **11.2P** – For ISPs with 2500s, 3600s, and 4000s in their backbone.²
- ✓ **11.2GS** – Current recommended release for ISPs with the 12000 series routers.
- ✓ **12.0S** – The new release for ISPs supporting the 7200, RSP7000, 7500 and 12000 series routers.

At the time of writing, many ISPs are running Early Deployment versions of 12.0S. Even though 11.1CC is very close to the end of its life (no features have been added since 11.1(26)CC), we still recommend 11.1CC as the best IOS version for ISPs. More and more ISPs are now planning migration to the 12.0S release as it includes support for some of the newer hardware and software features becoming available.

Cisco Systems’ most up to date recommendations on which IOS branch an ISP should be using will be on our Product Bulletin page available via CCO:

<http://www.cisco.com/warp/public/cc/cisco/mkt/gen/bulletin/>

² Yes, there are many ISPs in the world whose entire backbone is built on 2500s!

Saturday, February 19, 2000

Where to Get Information on 11.1CC?

11.1CC is available via CCO's Software Library. The following URLs have some additional details on the features included in 11.1CC, migration options, and how to download.

Cisco IOS Software Release 11.1CC New Features

http://www.cisco.com/warp/public/cc/cisco/mkt/ios/rel/111/prodlit/727_pb.htm

Cisco IOS Software Release 11.1CC Ordering Procedures and Platform Hardware Support

http://www.cisco.com/warp/public/cc/cisco/mkt/ios/rel/111/prodlit/728_pb.htm

Cisco IOS Software Release Process for Release 11.1 CC

http://www.cisco.com/warp/public/cc/cisco/mkt/ios/rel/111/prodlit/754_pp.htm

Cisco IOS 11.1CC Migration Guide

http://www.cisco.com/warp/public/cc/cisco/mkt/ios/rel/111/prodlit/111cc_dg.htm

Where to Get Information on 12.0S

12.0S is now available via CCO's Software Library <http://cco.cisco.com/kobayashi/sw-center/sw-ios.shtml>. The following URLs have some additional details on the features included in 12.0S, migration options, and how to download.

Cisco IOS Software Release 12.0S New Features

http://www.cisco.com/warp/public/cc/cisco/mkt/ios/rel/120/prodlit/934_pb.htm

Cisco IOS Software Release 12.0S Ordering Procedures and Platform Hardware Support

http://www.cisco.com/warp/public/cc/cisco/mkt/ios/rel/120/prodlit/935_pb.htm

Cisco IOS Software Release Notes for Release 12.0S

<http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/relnote/7000fam/rn120s.htm>

Cisco IOS 12.0S Migration Guide

http://www.cisco.com/warp/public/cc/cisco/mkt/ios/rel/120/prodlit/940_pb.htm

Further Reference on IOS Software Releases

Figure 1 provides a visual map of IOS releases and how the different versions and trains inter-relate. The following URLs on CCO will have more detailed and possibly up to date information on IOS release structure:

Cisco IOS Releases

<http://www.cisco.com/warp/public/732/Releases/>

Types of Cisco IOS Software Releases

http://www.cisco.com/warp/customer/cc/cisco/mkt/ios/rel/prodlit/537_pp.htm

Release Designations Defined - *Software Lifecycle Definitions*

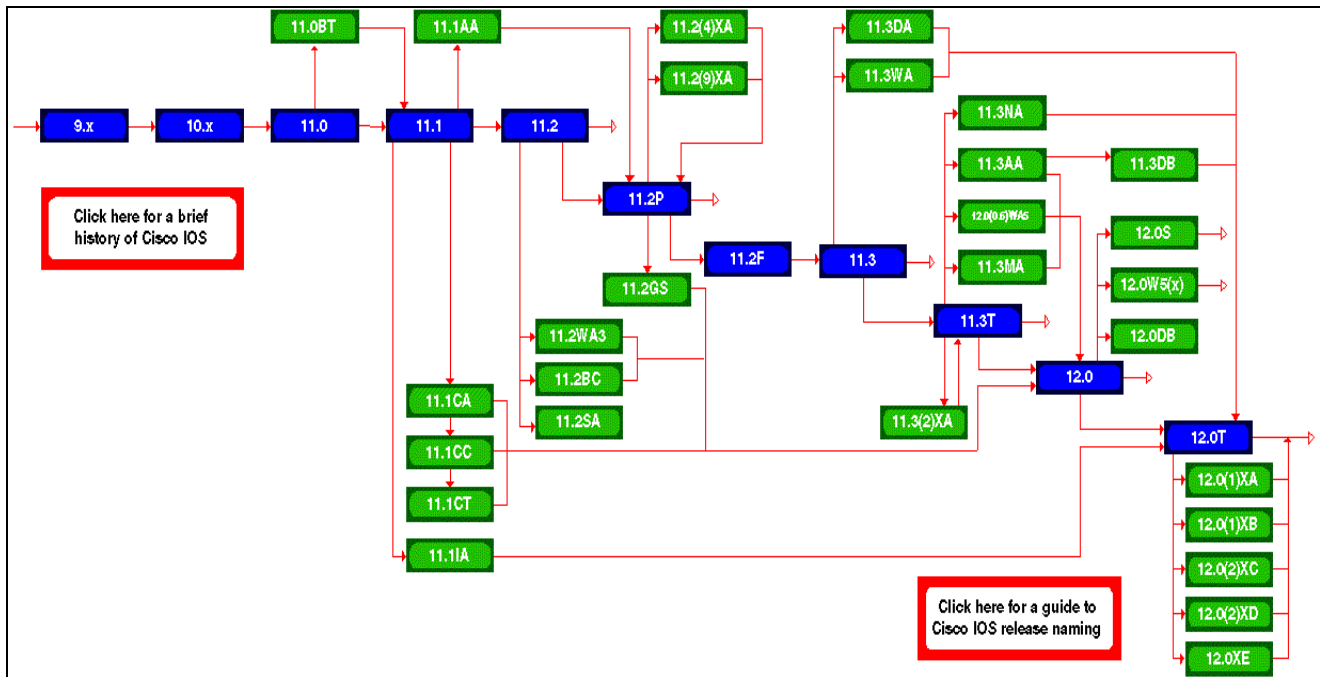
<http://www.cisco.com/warp/customer/417/109.html>

Software Naming Conventions for IOS

<http://www.cisco.com/warp/customer/432/7.html>

IOS Reference Guide

<http://www.cisco.com/warp/public/620/1.html>

Figure 1 – IOS Roadmap³

IOS Software Management

Most router platforms used in ISP backbone networks have very flexible RAM and FLASH memory configurations. For private, enterprise, or campus networks, the number of changes required in software, new features, or even the network infrastructure is small. The Internet is changing daily, growing daily, and a common mistake by new ISPs is to under specify the equipment they purchase. This should not be taken as a recommendation to buy what may seem like unneeded memory. It is recognition of the fact that having to upgrade a router every few months due to “unforeseen” growth in the Internet causes disruption to the network, and can potentially affect the reliability of hardware. Many Internet engineers support the view that “the more often humans touch a piece of hardware, the less reliable the hardware becomes”.

Flash Memory

The flash memory on a router is where the IOS software images are stored. When a new router is purchased, it will have the IOS image specified at time of ordering installed in flash. Flash memory is usually built into the router, and some platforms have expansion slots where PCMCIA flash cards can be installed.

It is good practice to supplement the built-in flash with another area of flash memory. This can be done in at least two ways:

1. Partition the built-in flash memory. This can be done using the configuration command, for example:

```
partition flash 2 8 8
```

which will partition the flash into two areas of 8Mbyte each (assuming 16Mbytes of installed flash memory, and the hardware supports this type of partitioning).

2. Install a flash card in one or both of the external flash slots.

³ Check <http://www.cisco.com/warp/public/620/roadmap.shtml> for updates to this roadmap.

Saturday, February 19, 2000

Having more than one flash memory area ensures that the router IOS image can be upgraded without affecting the existing image. For example, if there is only room for one image in flash, and it is the image the router is running, the existing image would have to be removed before a new one could be installed. What would happen, say, if the router crashed during the image upgrade? Recovery is possible with the boot image, but this is significantly harder than if proper precautions had been taken. By copying the new image into the other area of flash memory, the ISP ensures that the network functionality is minimally affected in the event of a crash or other unforeseen problems during image upgrade.

The new image in the second area of flash memory can easily be selected, as shown in the following example for the 7x00 series routers:

```
boot system flash slot1:rsp-pv-mz.120-5.S
boot system flash slot0:rsp-pv-mz.111-25.CC
boot system flash
```

which basically tells the router to boot rsp-pv-mz.120-5.S from slot1 flash first. If that image cannot be found or the flash card is not in slot1, it looks for rsp-pv-mz.111-25.CC in slot0. If that cannot be found the router boot software looks for the first image in any of the system flash.

Or this example, on the 36x0 series routers where the main 16Mb flash has been partitioned:

```
boot system flash flash:1:c3640-p-mz.120-5.S
boot system flash flash:2:c3640-p-mz.112-19.P
boot system flash
```

which basically tells the router to boot c3640-p-mz.120-5.S from the first flash partition. If the router cannot find that image, it will try to look for c3640-p-mz.112-19.P in the second flash partition. Failing that, it looks for the first usable IOS image in flash memory.

This type of arrangement ensures that in the event of image corruption, or a problem with the operating image, or router crash, there is always some backup image available on the router. Proper precautions ensure minimal network down time during maintenance periods or emergency occasions. Downloading a new image during a network down situation with customers and management exerting pressure is unnecessary extra stress which could easily have been avoided with a little precaution.

Common practice is for ISPs to leave the known working image on one of the flash cards or flash partitions of the router. In the event of deployment of a new release (which has passed tests in the lab environment) exhibiting some problem or defect later in operation, it is easy to back track to the old image.

Finally, it makes no commercial or operational sense to skimp on size of flash memory. As more and more of the features requested by ISPs are added to IOS, the image grows larger and larger. Sizing flash to the current image size is a false economy because it is more than likely that in the near future a larger image with new features may require flash memory larger than has been installed in the router.

System Memory

Another common practice amongst the Tier 1 and Tier 2 ISPs in all regions of the world is maximising the memory on every router. Cisco recommends the necessary amount of memory required to run each IOS image. Downloading a new image from CCO includes a question, which has to be answered, asking the user if they are fully aware of the memory requirements for the chosen image. Ignore the minimum recommendations at your peril!

For example, at the time of writing, it is recognised that 128Mbytes of memory is the minimum requirement to operate a router carrying the full Internet routing table. And any ISP requesting IOS 12.0S is required to acknowledge this fact. IOS 12.0S will operate inside 32Mbytes of memory, and will carry full Internet routes with 64Mbytes memory, but due to memory allocation issues, will not operate optimally. For example, the BGP algorithms will use more memory if it is available to improve their efficiency. Skimping on memory means affecting the performance of the routers, and the end result which the customer experiences.

Indeed many ISPs now simply specify maximum memory when they purchase new routing hardware. They recognise that sending an engineer to remove processor cards costs money through downtime, extra human resources, potential service disruption, and shortens the lifetime of the equipment through the human interaction. “Fit and Forget” is the norm amongst many of the largest ISPs today.

When and How to Upgrade?

Several ISPs upgrade their router software almost every time Cisco releases a new image. However, the only time any ISP should be upgrading software is when they require to fix bugs, support new hardware, or implement new software features. In many other industries, changing core operating software is seen as a major event, not undertaken lightly. Yet for some reason, some ISPs seem to think that a fortnightly upgrade is good practice.

Again, based on what most Tier 1 and Tier 2 ISPs now do, software upgrades are carried out only when they are required. Extensive testing is carried out in their test lab (how many ISPs have a test network which looks like one of their point’s of presence, or portion of their network?). Deployment only happens after extensive testing, and even then new images are implemented with caution on a quieter part of the network. Caution is of paramount importance on a commercial grade network. Even when upgrades are carried out, remember the recommendations above. IOS makes it easier by giving back-out paths via alternative images. Never attempt an upgrade without being aware of potential side-effects due to unforeseen problems; never attempt an upgrade without a back-out plan.

Another practice implemented by most Tier 1 and Tier 2 ISPs is to minimise the different versions of IOS images running on their network’s routers. The reason for this is almost always due to administrative and management reasons. Apart from reducing the number of potential interoperability issues due to bugs and new features, it is easier to train operations staff on the features and differences between few images than it is to train them on the differences between many images. Typically ISPs aim to run no more than two different IOS releases. One image is the old release, the other is the one which they are doing the blanket upgrade on their backbone. Upgrades tend to be phased, not carried en-masse overnight. If the ISPs have access equipment, such as the AS5x00 series, or the cable/xDSL aggregation devices they will deploy different IOS images on these devices. But again, if one dial box needs to be upgraded, ISPs tend to upgrade them all to ensure a consistent IOS release on that network.

Copying new images to FLASH Memory

Copying a new image into Flash memory in itself isn’t a complicated process, but there are a few good practice points to be aware of. The most important point here is to re-emphasise that leaving a back out image somewhere on the router is good practice and plain common sense. So many network outages have been prolonged because a new router image failed and the ISP hadn’t left a back out image on the device.

New images should be loaded into flash during maintenance periods, not when the router is live carrying full traffic load. While the activity of copying an image to flash won’t impact the router’s operation, it is advisable to avoid enhancing the possibility of an accident while the network is in production. At least an operational error during a maintenance period won’t cause customer anger as they were expecting downtime during the maintenance period in any case (assuming the customers were informed in the first place, another key point several ISPs seem to forget!).

The commands to copy software into flash memory have been refined in releases from 12.0, making the mechanics of getting software to and from the router simpler and more consistent in style. The “copy” command has been enhanced to support a URL appearance covering all system devices in a consistent format, for example as:

```
beta7200#copy tftp ?
bootflash:      Copy to bootflash: file system
disk0:          Copy to disk0: file system
disk1:          Copy to disk1: file system
flash:          Copy to flash: file system
ftp:            Copy to ftp: file system
lex:            Copy to lex: file system
null:           Copy to null: file system
nvram:          Copy to nvram: file system
```

Saturday, February 19, 2000

```
rcp:          Copy to rcp: file system
running-config Update (merge with) current system configuration
slot0:       Copy to slot0: file system
slot1:       Copy to slot1: file system
startup-config Copy to startup configuration
system:      Copy to system: file system
tftp:        Copy to tftp: file system
```

This is somewhat improved over the rather inconsistent if platform dependent format used in previous releases.

Before copying the image to flash, make sure the flash has enough space for the new image. Preferably install two flash devices, or partition the existing flash device, as has been described previously. Use “cd”, “delete”, “erase” and “squeeze” (depending on platform) to clear enough space on the flash file system. Make sure that the partition/flash holding the currently running image is not touched. If there is any problem with the upgrade, a back out path is available. And don’t forget to set up the “boot system xxxxx” IOS configuration command so that the router is told to boot the currently running image.

Once the flash partition is ready, a copy command can be issued to copy the new IOS image from the remote device (could be any of those listed above) to the partition. An example of the copy command is given below:

```
beta7200#copy tftp slot1:
Address or name of remote host []? noc1
Source filename []? 12.0S/c7200-p-mz.120-6.S
Destination filename [c7200-p-mz.120-6.S]?
Accessing tftp://noc1/12.0S/c7200-p-mz.120-6.S...
Loading 12.0S/c7200-p-mz.120-6.S from 192.168.3.1 (via Serial3/1):
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 5708964/11417600 bytes]

5708964 bytes copied in 330.224 secs (17299 bytes/sec)
beta7200#
```

This will copy the image c7200-p-mz.120-6.S from the tftp server to the flash device in slot1. Or the command can be shortened to:

```
beta7200#copy tftp://noc1/12.0S/c7200-p-mz.120-6.S slot1:
Destination filename [c7200-p-mz.120-6.S]?
...etc...
```

which will achieve the same thing. Notice that the router will attempt to work out the file name from the URL string entered – this can be helpful and save typing. When successfully verified, set up the router to boot the new image (use the “no boot system” and “boot system” commands described earlier) – it is also a good idea to configure another boot system command pointing to the backup image (as in the example in the earlier section). If desirable, the router can even be configured to do a timed/delayed reboot at some time in the future. Use that feature with care though; it is perfectly feasible to do timed reboots on several routers and completely break a portion of the ISP network! An example:

```
beta7200# reload at 17:05
```

will reboot the router at 17:05 local time.

When the new image has been booted successfully, it should be put through acceptance tests during the maintenance slot (could be as simple as “does the routing work?”, for example), and then monitored during operation. Don’t delete the previous image – you know it works so if it is left on the other flash partition a back out path is available in case of future problems. The old can be deleted once a decision has been made to further upgrade the IOS. The benefit of configuring two flash devices/partitions is clear to see – ease of maintenance!

Summary: upgrade only when bug fixes, or new hardware support, or new software features are required. Otherwise, do not touch that router! “If it isn’t broken, don’t fix it.”

Configuration Management

NVRAM and TFTPserver

The onboard router NVRAM is used to store the router's active configuration. Most ISPs keep an off-router copy of this configuration too. In the unlikely event of the configuration being lost on the router, they can quickly recover the system with the off-router backup. There are several options for off-router backup of the running configuration:

- Write configuration to a tftp server using the “write net” command. (In 12.0 and more recent software the “write net” command has been superseded with the more sophisticated copy function. The equivalent command is “copy running tftp:”.)
- Configurations saved by an operator's “write net” command are kept under automated revision control. Combined with TACACS+ authentication (see later) it is possible to track whom has changed what and when. Important for accountability and configuration back-out in case of problems.
- An automated (e.g. Unix cron) job collects the configuration from the router every few hours or so. The collected copy is kept under revision control. Changes could be flagged to the network monitoring system, or to the NOC, or to the operations engineers.
- Router configurations are built from a master configuration database. The running configuration is only a copy, with the master configuration kept off-router. Updates to the running configuration are made by altering the master files (under revision control) and implementing the new master configuration during maintenance periods.

Note: See the Chapter discussing loopback interfaces for best practice for configuring the router for tftp services.

The IOS command prompts to save the configuration are given in the following examples. The syntax has been significantly changed starting with IOS 12.0, mainly to make the commands used to transfer configuration files and IOS software between operator/NOC and the router more consistent. The IOS command prior to 12.0 is given in the following example:

```
alpha7200#write network
Remote host []? noc-pc
Name of configuration file to write [alpha7200-config]?
Write file router2-config on host 192.168.3.1? [confirm]
Building configuration...

Writing alpha7200-config !!! [OK]
alpha7200#
```

From 12.0 onwards, the command to do the same thing is *copy* as given in the following example:

```
beta7200#copy running tftp:
Remote host[]? noc-pc
Destination filename [beta7200-config]?
Write file tftp://noc-pc/beta7200-config? [confirm]
!!! [OK]
beta7200#
```

Note: the “write network” command is still supported in 12.0, although may be withdrawn in a future release.

Large Configurations

When the NVRAM is not large enough to store the router configuration there is an option which allows the configuration to be compressed (using a gzip like compression algorithm):

```
service compress-config
```

Only use this if there is a requirement to. If the existing NVRAM can hold the configuration uncompressed, do not use this feature. Some ISPs have extremely large configurations and this feature was introduced to assist them.

Saturday, February 19, 2000

Furthermore, if the router configuration has become very large it is worth checking whether some of the newer IOS features can be used. One example would be using prefix-lists instead of access-lists; the former is more space efficient in NVRAM, and also is more efficient in operation.

Detailed Logging

Keeping logs is a common and accepted operation practice. Interface status, security alerts, environmental conditions, CPU process hog and many other events on the router can be captured and analysed via UNIX syslog. Cisco System's IOS has the capability to do UNIX logging to a UNIX syslog server. Cisco System's UNIX syslog format is compatible with 4.3 BSD UNIX. The follow is a typical logging configuration for ISPs:

```
logging buffered 16384          <-- 16Kbyte history buffer on router
logging trap debugging          <-- Syslog server logging level set to debug
logging facility local7         <-- Syslog facility on syslog server
logging 169.223.32.1           <-- IP address of your first syslog server
logging 169.223.45.8           <-- IP address of your second syslog server
```

To set up the syslog daemon on a 4.3 BSD UNIX system, include a line such as the following in the file /etc/syslog.conf:

```
local7.debugging      /var/log/cisco.log
```

Notice that it is considered good practice to reserve a syslog facility on the Unix log host for each type of network device. So, for example, backbone routers may use "local7", access servers may use "local5", tacacs+ may use "local3", etc. Putting all the logs in one huge file simply makes system management hard, and debugging problems by searching the log file next to impossible. Notice that the more modern Unix platforms might require network support in the syslog daemon to be enabled by a runtime option (network support is now disabled by default to avoid security problems).

By default, log messages are not time stamped. If you do configure your routers for UNIX logging, you will want detailed timestamps of for each log entry:

```
service timestamps debug datetime localtime show-timezone msec
service timestamps log datetime localtime show-timezone msec
```

which will produce a syslog message looking something similar to:

```
Jul 27 15:53:23.235 AEST: %SYS-5-CONFIG_I: Configured from console by philip on console
```

The command line options in the timestamps command are as follows:

- debug: all debug info is time stamped
- log: all log info is time stamped
- datetime: the date and time is include in the syslog message
- localtime: the local time is used in the log message (as opposed to UTC)
- show-timezone: the timezone defined on the router is included (useful if the network crosses multiple time-zones)
- msec: time accuracy to milliseconds – useful if NTP is configured.

By default, a syslog message contains the IP address of the interface it uses to leave the router. You can require all syslog messages to contain the same IP address, regardless of which interface they use. Many ISPs use the loopback IP address. This keeps their syslogs consistent plus allows them to enhance the security of their SYSLOG server host (by the use of TCP wrappers or router filters for example).

```
logging source-interface loopback0
```

Note: See the Chapter discussing loopback interfaces for best practice for configuring the log hosts for syslog services.

Analysing Syslog Data

Configuring the routers to export syslog data is one step. The next step is to take the data, store it, analyse it, and use it in the day to day operations. Interface status, security alerts, and debugging problems are some of the most common events ISPs monitor from the collected syslog data. Some use custom written PERL scripts to create simple reports. Others use more sophisticated software to analyse the syslog data and create html reports, graphs, and charts.

The following is a list of known available software that analyses syslog data. Even if you are going to write your own scripts, it's worth checking out the commercial packages to see what can be done with syslog data.

Cisco Resource Manager	http://www.cisco.com/warp/public/cc/cisco/mkt/enm/rman/index.shtml
Private I	http://www.4privatei.com/
Crystal Reports	http://www.seagatesoftware.com/crystalreports/

Network Time Protocol (NTP)

NTP is *THE* most overlooked feature on an ISP's network. If you wish to compare the syslog information from devices all over your network, you will need to synchronise the time on all the routers. Comparing logs from various networks is essential for many types of troubleshooting, fault analysis, and security incident tracking. Without precise time synchronisation between all the various logging, management, and AAA functions, this sort of comparison would be impossible.

The Network Time Protocol (NTP) is a protocol designed to synchronise the clocks on a network of computing and communication equipment. NTP runs over UDP, which in turn runs over IP. NTP implements a version of the Network Time Protocol first described in RFC-958, "Network Time Protocol". Other RFCs for time synchronisation include:

- RFC-1119, "Network Time Protocol (Version 2) Specification and Implementation", 1989 (obsoletes RFC-1059, RFC-958).
- RFC-1128, "Measured Performance of the Network Time Protocol in the Internet System", 1989.
- RFC-1129, "Internet Time Synchronisation: The Network Time Protocol", 1989.
- RFC-1165, "Network Time Protocol (NTP) over the OSI Remote Operations Service", 1990.
- RFC-1305, "Network Time Protocol", 1992.

An NTP network usually gets its time from an authoritative time source, such as a radio clock, or a Global Positioning System (GPS) device, or an atomic clock attached to a timeserver. NTP then distributes this time across the network. NTP is extremely efficient; no more than one packet per minute is necessary to synchronise two machines to within a millisecond of one another.

NTP Architecture⁴

In the NTP model, a number of primary reference sources, synchronised by wire, GPS, or radio to national standards, are connected to widely accessible resources, such as backbone gateways, and operated as primary time servers. NTP provides a protocol to pass timekeeping information from these servers to other time servers via the Internet and to cross-check clocks and correct errors arising from equipment or propagation failures. Local-net hosts or gateways, acting as secondary time servers, use NTP to communicate with one or more of the primary servers. In order to reduce the protocol overhead, the secondary servers distribute time to the remaining local-net hosts. For reliability, selected hosts are equipped with less accurate (and less expensive) radio clocks. These hosts are used for backup in case of failure of the primary and/or secondary servers or the communication paths between them.

⁴ This section was written for Cisco's DNS/DHCP Manager. Sections of the documentation on NTP have been included in this document. The complete document can be found at: <http://www.cisco.com/univercd/cc/td/doc/product/iaabu/cddm/cddm111/adguide/ntp.htm>

Saturday, February 19, 2000

The NTP “network” consists of a multiple redundant hierarchy of servers and clients, with each level in the hierarchy identified by a stratum number. This number specifies the accuracy of each server, with the topmost level (primary servers) assigned as 1 and each level downward (secondary servers) in the hierarchy assigned as one greater than the preceding level. Stratum 1 is populated with hosts with bus or serial interfaces to reliable sources of time, such as radio clocks, GPS satellite timing receivers, or atomic clocks. Stratum 2 servers might be company or campus servers that obtain time from some number of primary servers over Internet paths, and provide time to many local clients. The stratum 2 servers may be configured to peer with each other, comparing clocks and generating a synchronised time value.

NTP performs well over the non-deterministic path lengths of packet-switched networks, because it makes robust estimates of three key variables in the relationship between a client and a time server. These three variables are: network delay, dispersion of time packet exchanges (a measure of maximum clock error between the two hosts), and clock offset (the correction to apply to a client's clock to synchronise it). Clock synchronisation at the 10-millisecond level over long distance (2000 km) WANs, and at the 1-millisecond level for LANs, is routinely achieved.

There is no provision for peer discovery or virtual-circuit management in NTP. Data integrity is provided by the IP and UDP checksums. No flow-control or retransmission facilities are provided or necessary. Duplicate detection is inherent in the processing algorithms.

NTP uses a system call on the local host to “slew” the local system clock by a small amount in order to keep the clock synchronised. If the local clock exceeds the “correct” time by pre-set threshold, then NTP uses a system call to make a step adjustment of the local clock.

NTP is careful to avoid synchronising to a system whose time may not be accurate. It avoids doing so in two ways. First of all, NTP will never synchronise to a system that is not itself synchronised. Secondly, NTP will compare the time reported by several systems, and will not synchronise with a system whose time is significantly different from the others, even if its stratum is lower.

Client/Server Models and Association Modes

There are a number of modes in which NTP servers can associate with each other. The mode of each server in the pair indicates the behaviour the other server can expect from it. An “association” is formed when two peers exchange messages and one or both of them create and maintain an instantiation of the protocol machine. The association can operate in one of several modes: server, client, peer, and broadcast/multicast. The modes are further classified as active and passive. In active modes, the host continues to send NTP messages regardless of the reachability or stratum of its peer. In passive modes, the host sends NTP messages only as long as its peer is reachable and operating at a stratum level less than or equal to the host; otherwise, the peer association is dissolved.

- **Server Mode** – By operating in server mode, a host (usually a LAN time server) announces its willingness to synchronise, but not to be synchronised by a peer. This type of association is ordinarily created upon arrival of a client request message and exists only in order to reply to that request, after which the association is dissolved. Server mode is a passive mode.
- **Client Mode** – By operating in client mode, the host (usually a LAN workstation) announces its willingness to be synchronised by, but not to synchronise the peer. A host operating in client mode sends periodic messages regardless of the reachability or stratum of its peer. Client mode is an active mode.
- **Peer Mode** – By operating in peer mode (also called “symmetric” mode), a host announces its willingness to synchronise and be synchronised by other peers. Peers can be configured as active (symmetric-active) or passive (symmetric-passive).
- **Broadcast/Multicast Mode** – By operating in broadcast or multicast mode, the host (usually a LAN time server operating on a high-speed broadcast medium) announces its willingness to synchronise all of the peers, but not to be synchronised by any of them. Broadcast mode requires a broadcast server on the same subnet, while multicast

mode requires support for IP multicast on the client machine, as well as connectivity via the MBONE to a multicast server. Broadcast and multicast modes are active modes.

Normally, one peer operates in an active mode (symmetric-active, client or broadcast/multicast modes), while the other operates in a passive mode (symmetric-passive or server modes), often without prior configuration. However, both peers can be configured to operate in the symmetric-active mode. An error condition results when both peers operate in the same mode, except for the case of symmetric-active mode. In this case, each peer ignores messages from the other, so that prior associations, if any, will be demobilised due to reachability failure.

Implementing NTP on an ISP's Routers

The time kept on a machine is a critical resource, so we strongly recommend that you use the security features of NTP to avoid the accidental or malicious setting of incorrect time. Two mechanisms are available: an access list-based restriction scheme and an encrypted authentication mechanism. The example below highlights both NTP security options.

Cisco's implementation of NTP does not support stratum 1 service; in other words, it is not possible to connect a router running IOS directly to a radio or atomic clock. It is recommended that time service for your network is derived from the public NTP servers available in the Internet. If the network is isolated from the Internet, Cisco's implementation of NTP allows a system to be configured so that it acts as though it is synchronised via NTP, when in fact it has determined the time using other means. Other systems then synchronise to that system via NTP. The command to set up a router in this way is:

```
ntp master 1
```

which tells the router that it is the master time source, and running at Stratum 1.

The example below is a NTP configuration on a router getting a Stratum 2 server connection from 192.36.143.150 and peering with 169.223.50.14. The peered IP addresses are the loopback addresses on each router. Each router is using the loopback as the source. This makes security easier (note the access-list).

```
clock timezone SST 8
!
access-list 5 permit 192.36.143.150
access-list 5 permit 169.223.50.14
access-list 5 deny any
!
ntp authentication-key 1234 md5 104D000A0618 7
ntp authenticate
ntp trusted-key 1234
ntp source Loopback0
ntp access-group peer 5
ntp update-calendar
ntp server 192.36.143.150
ntp peer 169.223.50.14
!
```

NTP Deployment Examples

ISPs use several designs to deploy NTP on their backbones. It is hard to recommend a best current practice, but this section lists a few examples:

- Flat peer structure. Here all the routers peer with each other, with a few geographically separate routers configured to point to external systems. From experience this is very stable, but start up will take a very long time if being deployed on a live running network.
- Hierarchy. Here the BGP route reflector hierarchy is "copied" for the NTP hierarchy. Core routers (route reflectors) have a client/server relationship with external time sources, the reflector clients have a client/server relationship with the core routers, the customer routers have a client/server relationship with the reflector clients,... And so on down the tree. A simple hierarchy which seems to be very stable also.

Saturday, February 19, 2000

- External. Here all the ISP routers have a client/server relationship with a few time “servers” in the ISP’s backbone, usually Unix systems synchronised with external time sources, or their own GPS receiver. This set up is also reported to be very stable.

Undoubtedly there are other possibilities too. The main aim is to go for stability as time synchronisation is a key tool within the ISP backbone.

Further NTP References

Here are some URLs with further pointers to NTP information, software, and hardware:

The Network Time Protocol (NTP) Master Clock	http://tycho.usno.navy.mil/
Datum Inc, Bancomm Timing Division	http://www.datum.com/
The Time of Internet - Network Time Protocol	http://www.cstv.to.cnr.it/toi/uk/ntp.html
The Time Web Server (Time Sync) by Dave Mills	http://www.eecis.udel.edu/~ntp/
Coetanian Systems Time Synchronisation Server 100	http://www.coetanian.com/tss/tss100.htm

Simple Network Management Protocol (SNMP)

Keeping data on the health of an ISP’s network is critical to its survival as a business. For example, an ISP must know the load on their backbone circuits, and the loading on their customer circuits. They also need to keep track of the packets lost on routers at various points of the network. And they need to be aware of long term trends on the overall growth of the network. Simple Network Management Protocol (SNMP) can collect and process all of this data – hence it is a very critical utility for Network Engineers. Given the wide range of freeware, shareware, and commercially available SNMP tools, all ISPs should be able to collect SNMP data, process it, graph it, and analysis it for proper traffic engineering. Appendix 3 – Traffic Engineering Tools – lists pointers to various software and tools on the Net.

Yet, remember that SNMP, especially version 1, has very weak security! If SNMP is not going to be used, turn it off! On the other hand, ensure that there is sufficient configuration information present which will control the use of SNMP such that it doesn’t become a security risk. Most importantly, never leave a configuration which includes “public” or “private” as the community string – these strings are so well known, and are common defaults on hardware shipping from so many vendors that they are open invitations to abuse, filters or not.

If SNMP is used in a read-only scenario, ensure that it is set up with appropriate access controls. The following is an example:

```
access-list 98 permit 215.17.34.1
access-list 98 permit 215.17.1.1
access-list 98 deny any
snmp-server community 5nmc02m RO 98
snmp-server trap-source Loopback0
snmp-server trap-authentication
snmp-server enable traps config
snmp-server enable traps envmon
snmp-server enable traps bgp
snmp-server enable traps frame-relay
snmp-server contact Barry Raveendran Greene [bgreene@cisco.com]
snmp-server location Core Router #1 in City Y
snmp-server host 215.17.34.1 5nmc02m
snmp-server host 215.17.1.1 5nmc02m
snmp-server tftp-server-list 98
!
```

Note the application of *access-list 98* above. The community string *5nmc02m* is not encrypted, hence the need to use an access-list to control access. This is too often forgotten, and if the community string is known outside the ISP, it can easily

lead to compromise of a router. In fact, there are scripts available on the Internet which allow *script kiddies*⁵ to probe a router and crack the community name. Unless BOTH SNMP is set up to send a trap on community name authentication failure AND the SNMP management device is configured to react to the authentication failure trap, the *script kiddies* will most likely discover the community name.

ISPs should always remember that accepting SNMP only from “known good” IP addresses does not guarantee the security. Unless the ISP has some very serious anti-spoofing measures in place, you **cannot** completely rely on IP addresses for the primary security of any system. IP addresses are frequently spoofed. Layered security where the system relies on several mechanisms has proven to be more effective.

The *snmp-server host* configuration lists the hosts to which SNMP information is sent – if there is no means of collecting SNMP traps, don’t configure *snmp-server host*, saving CPU cycles and network bandwidth. ISPs should ensure that the *snmp-server host* is configured to receive and respond to SNMP traps. For instance, a PERL script on a PC running Unix or Linux with UCD SNMP⁶ could receive an SNMP environmental trap relating to high router internal temperature, e-mail it to the NOC alias, send an alphanumeric page to the on-duty engineer, and open a trouble ticket.

If SNMP is going to be used in read/write mode, think very very carefully about the configuration and why there is a requirement to do this, as configuration errors in this scenario could leave the router very vulnerable.

If possible, put an ACL at the edge of your network that will prevent outside parties from probing your network via SNMP. There are many publicly and commercially available tools which will scan ANY network on the Internet via SNMP. This could map out your entire network and/or discover a device that has had SNMP left open.

HTTP Server

The *http* server is a new feature in 11.1CC and 12.0 software which, when configured and enabled, allows the network operator to view and configure the router through a convenient and easy to use Web interface (via common browsers such as Netscape or Internet Explorer). If there is no intention of using the built-in HTTP server in the running configuration, it is worth checking that it has not been enabled by default, or in error, or during system installation. The configuration command:

```
no ip http server
```

will ensure that the server is not running.

If there is a need to configure the *http* server because Web based configuration of the router is desired or desirable, then we strongly advise that the server is configured with the appropriate security. For example:

```
ip http server
ip http port 8765                ! use a non-standard port
ip http authentication aaa       ! use the AAA authentication method which has been configured
ip http access-class <1-99>     ! access-list to protect the HTTP port
```

Notice the suggestion of a non-standard port. This adds a little obscurity to the Web server on the router, making potential attack a little more difficult. Also notice the access-list used, and the authentication type (AAA is discussed in the section on router security). This ensures that only the permitted administrative users of the router get access to the device from the authorised IP address range.

⁵ Script Kiddies are amateur crackers who use scripts to break into and cause damage to networks and systems on the Internet.

⁶ CMU SNMP has not been updated in a while and the project has now been taken over by UCD. UCD SNMP contains a port and modified code of the CMU 2.1.2.1 snmp agent. It has been modified to allow extensibility quickly and easily. It is by far the best and most configurable system – and it’s free!

<ftp://ucd-snmp.ucdavis.edu/> for the source

<http://ucd-snmp.ucdavis.edu/> for the UCD-SNMP project home page

Saturday, February 19, 2000

Core Dumps

A *core dump* facility has been part of IOS for several years and many software releases. The core dump facility operates like the Unix variant – when a programme crashes, the memory image is stored in a “core” file. When a router crashes, a copy of the core memory is kept. Before the memory is erased on reboot, the router can be set up to copy the core dump out to a UNIX server. An account (ftp, tftp, or rcp) and sufficient disk space (equal to the amount of memory on the router per dump) needs to be set up and allocated.

Here is an example using ftp:

```
ip ftp source-interface Loopback0
ip ftp username cisco
ip ftp password 7 045802150C2E
exception protocol ftp
exception dump 169.223.32.1
```

Note the use of the loopback interface as a source interface. It is recommended that access to the “cisco” account above be made as secure as possible. For example, do not send core dumps to the same ftp server as the one used to provide generic anonymous or user ftp accounts. Use a wrapper for the ftp daemon, and make sure that only the loopback interfaces are listed in any system filter lists.

Be aware that rcp is inherently insecure and its use cannot be recommended over a public network. Also tftp core dumps (which are the default in IOS) only support system memory sizes up to 16Mbytes. Generally it is recommended that ftp core dumps be configured whatever the situation, or router hardware configuration.

More detailed information for configuring core dumps on a Cisco IOS based system is located on the Cisco Documentation CD. It is publicly available via the Web at:

Creating Core Dumps – http://www.cisco.com/univercd/cc/td/doc/cisintwk/itg_v1/itga_cor.htm

It includes information needed to troubleshoot problems using the *core dump* and *show stacks* commands.

GENERAL FEATURES

This section covers general features which ISPs should consider for their routers and network implementations. Most are good design practices rather than leveraging particular unique IOS features, but each demonstrates how IOS can aid the smooth operation of an ISP's business.

Command Line Interface

The IOS Command Line Interface (CLI) is the traditional (and favoured) way of interacting with the router, to enter and change configuration, and to monitor the router's operation. The CLI is now very well documented in the Cisco UniverCD documentation set, e.g. at http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/fun_r/index.htm. There are, however, a few tips and tricks which are regularly used by ISPs and are worth mentioning here.

Editing Keys

There are several keys which are very useful for editing the IOS configuration. While these are documented in detail in the IOS 12.0 documentation set, it is useful to point out the most commonly used here.

- The TAB key while typing in an IOS configuration command will complete the command being typed in. This saves typing effort and is especially useful when the operator is still learning the IOS command set.
- The “?” key while typing in an IOS configuration command will list the available commands starting with the characters entered so far.
- Up arrow and down arrow respectively allows the operator to scroll up and down through the history buffer.
- CTRL-A moves the cursor to the beginning of the line
- CTRL-E moves the cursor to the end of the command line
- CTRL-K deletes all characters from the cursor to the end of the command line
- CTRL-W deletes the word to the left of the cursor
- CTRL-X deletes all characters from the cursor to the beginning of the command line
- ESC-B moves the cursor back one word
- ESC-F moves the cursor forward one word

The complete list of commands can be found in the IOS documentation:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/fun_r/frprt1/frui.htm

CLI String Search

A new feature in IOS from 11.1CC and 12.0 is a Unix grep-like function allowing operators to search for common expressions in configuration and other terminal output. See the following IOS document for complete information:

<http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120t/120t1/cliparse.htm>

The function is invoked by using a vertical bar “|”, like the Unix “pipe” command. For example:

```
beta7200#show configuration | ?
begin      Begin with the line that matches
exclude    Exclude lines that match
include    Include lines that match
```

Following one of these three options the operator should enter a regular expression to get a pattern match on the configuration, as in this example above. The regular expressions can be single or multiple characters or a more complex construction, in a similar style to Unix regular expressions.

Saturday, February 19, 2000

During the displaying of configuration or file contents, the screen pager “—More—“ will be displayed if the output is longer than the current terminal length setting. It is possible to do a regular expression search at this prompt too. The “/” key matches the “begin” keyword above, the “-” key means exclude, and the “+” key means include.

Finally, in enable mode, it is possible to use the “more” command to display file contents. Regular expressions as in the example shown above can be used with “more”. The options available with more are given in this example:

```
beta7200#more ?
 /ascii      Display binary files in ascii
 /binary     Force display to hex/text format
 /ebcdic     Display binary files in ebcdic
 bootflash:  File to display
 disk0:      File to display
 disk1:      File to display
 flash:      File to display
 ftp:        File to display
 null:       File to display
 nvram:      File to display
 rcp:        File to display
 slot0:      File to display
 slot1:      File to display
 system:     File to display
 tftp:       File to display
```

By using the “|” after the more command and its option above, it is possible to search within the file for the strings of interest in the same way as discussed previously.

Interface Configuration

Configuring interfaces is more than simply plugging in the cable and activating the interface with the IOS command “no shutdown”. Attention should be applied to details such as whether it is WAN or LAN, a routing protocol is running across the interface, addressing and masks to be used, and operator information.

Description

Use the “*description*” interface command to document details such as the circuit bandwidth, the customer name, the database entry mnemonic, the circuit number the circuit supplier gave you, and the cable number. Sounds like overkill, especially if there is a customer database within the ISP organisation. However, it is very easy to pick up all the relevant details from the router “*show interface*” command if and when an engineer needs to be on site, or is away from the database system, or the database happens to be unavailable. There can never be too little documentation, and documentation such as this ensures that reconstructing configurations and diagnosing problems are made considerably easier.

Bandwidth

Don’t forget the “*bandwidth*” interface command. It is used by interior routing protocols to decide optimum routing and especially important to be set properly in the case of backbone links using only a portion of the available bandwidth support by the interface. For example a serial interface (Serial0/0) on a router supports speeds up to 4Mbps, but has a default bandwidth setting of 1.5Mbps. If the backbone has different size links from 64Kbps to 4Mbps and the bandwidth command is not used, the interior routing protocol will assume that all the links have the same cost, and calculate optimum paths accordingly – this may be less than ideal.

On customer links it may seem that this setting is superfluous as an interior routing protocol is never run over a link to a customer. However, it never the less provides very useful online documentation for what the circuit bandwidth is. Furthermore, the bandwidth on the circuit is used to calculate the interface load variable – some ISPs monitor their

customer interfaces loading by SNMP polls so that they can get prior warning of problems, or congestion, or proactively inform customers of necessary upgrades. (Some ISPs look at the load variable, other ISPs look at the 5 minute average, inbound and outbound. If you monitor the load variable, you need to set the bandwidth so that it matches the true circuit bandwidth, not the default configured on the router.)

IP Unnumbered

Traditionally ISPs have used IP addresses for the point to point links on leased line circuits to customers. Indeed, several years ago, prior to the advent of CIDR, it was not uncommon to see a /26 or even a /24 used for simple point to point link addresses. With the advent of CIDR, /30 networks have been used instead (/30 is a block of four addresses, two of which can be used for physical interfaces). However, this has started to lead to problems too as IGP's of some of the larger ISPs are starting to carry several thousand networks, affecting convergence time, and becoming an administrative and documentation nightmare.

To avoid problems with large numbers of /30s floating around the ISP's internal routing protocol, and avoid the problems of keeping internal documentation consistent with network deployment (especially true in larger ISPs), many are now using "unnumbered" point to point links.

An unnumbered point to point link is one which requires no IP addresses. The configuration is such that traffic destined for one network from another is simply pointed at the serial interface concerned. "*ip unnumbered*" is an essential feature applicable to point-to-point interfaces such as Serial, HSSI, POS, etc. It allows the use of a fixed link (usually from ISP to customer) without consuming the usual /30 of address space, thereby keeping the number of networks routed by the IGP low. The "*ip unnumbered*" directive specifies that the point-to-point link should use an address of another interface on the router, typically a LAN, or more usually a Loopback interface. Any networks, which require to be routed to the customer, are pointed at the serial interface rather than the remote address of the point-to-point link, as would be done in normal instances.

Caveats

There are some situations which ISPs need to consider before implementing an IP unnumbered system for their customer point to point connections. These are considerations only – bear in mind that many ISPs have used IP unnumbered for several years, mainly so that they can control the size of the IGP running in their backbone network.

- **Pinging the Customer.** Many ISPs use monitoring systems which use "ping" to check the status of the leased line (customer connectivity). Even if the customer unplugs the LAN, an alarm will not be raised on the ISP's management system. This is because the customer router still knows that the LAN IP address is configured on the system and "useable". So long as the IP address is configured on the LAN there will be no reachability issues with using "ip unnumbered".
- **Routing Protocols.** If a routing protocol needs to be run over this link, it is operationally much easier to use IP addresses. Don't use "*ip unnumbered*" if the customer is peering with you using BGP across the link, or if the link is an internal backbone link. Simply use a network with a /30 address mask. (Routing will work over unnumbered links but the extra management and operational complexity probably outweighs the small address space advantage gained.)
- **Loopback Interfaces on the Customer's Router.** These offer no advantage to addressing the "ping" problem, and unnecessarily consume address space (not to mention adding complexity to the customer router configuration).

A Full Example

Using the above configuration commands, a typical configuration on the ISP's router would be as follows:

```
interface loopback 0
  description Loopback interface on Gateway Router 2
  ip address 215.17.3.1 255.255.255.255
  no ip redirects
  no ip directed-broadcast
```

Saturday, February 19, 2000

```
no ip proxy-arp
!
interface Serial 5/0
description 128K HDLC link to Galaxy Publications Ltd [galpub1] WT50314E R5-0
bandwidth 128
ip unnumbered loopback 0
no ip redirects
no ip directed-broadcast
no ip proxy-arp
!
ip route 215.34.10.0 255.255.252.0 Serial 5/0
```

The customer router configuration would look something like:

```
interface Ethernet 0
description Galaxy Publications LAN
ip address 215.34.10.1 255.255.252.0
no ip redirects
no ip directed-broadcast
no ip proxy-arp
!
interface Serial 0
description 128K HDLC link to Galaxy Internet Inc WT50314E C0
bandwidth 128
ip unnumbered ethernet 0
no ip redirects
no ip directed-broadcast
no ip proxy-arp
!
ip route 0.0.0.0 0.0.0.0 Serial 0
```

In this example the Regional or Local Registry has allocated the customer the network block 215.34.10.0/22. This is routed to the customer site with the static route pointing to Serial 5/0 above. The customer router simply needs a default route pointing to its serial interface to ensure a connection.

With this configuration, there are no /30s from point to point links present in the IGP, and the ISP does not need to document the link address, or keep a table/database up to date. It all makes for easier configuration, and easier operation of the ISP's business.

Note that the loopback interface used for the ISP's router is the same loopback interface which would be used for the iBGP. There is no advantage to configuring a second loopback interface. Note the contents of the description field. This example has included:

- the bandwidth of the circuit, **128K**
- the encapsulation, **HDLC**
- the name of the company, **Galaxy Publications Ltd**
- the database mnemonic in the ISP's internal database, **[galpub1]**
- the telco's circuit ID, **WT50314E**
- the cable number **R5-0**

All of these are online documentation, seemingly superfluous, but so necessary/essential to ensure smooth and efficient operations. All the information pertinent to the customer's connection from the cabling to the IP values is contained in the interface configuration. Should the ISP's database be down, or unavailable, any debug information required by operators or engineers can be found on the router itself.

NetFlow

Enabling NetFlow on routers provides network administrators with access to "packet flow" information from their network. Exported NetFlow data can be used for a variety of purposes, including security monitoring, network management and planning, customer billing, and Internet traffic flow analysis.

NetFlow is available on all router platforms from the 2600 series upwards from the 12.0 software release onwards. It was first introduced in 11.1CC on the 7200 and 7500 platforms. It can be enabled on a per-interface basis on the routers as in the following example:

```
interface serial 5/0
 ip route-cache flow
!
```

If CEF is not configured on the router, this will turn off the existing switching path on the router and enable NetFlow switching (basically modified optimum switching). If CEF is configured on the router, NetFlow simply becomes a “flow information gatherer” – CEF remains operational as the underlying switching process.

To view NetFlow information on the router, simply enter the command “show ip cache flow”. This will display the current flow cache on the terminal screen. An example might be:

```
gw.test.int>sh ip cache flow
IP packet size distribution (437938623 total packets):
 1-32  64  96 128 160 192 224 256 288 320 352 384 416 448 480
 .000 .059 .527 .007 .258 .063 .049 .007 .001 .012 .000 .000 .000 .000 .000

 512  544  576 1024 1536 2048 2560 3072 3584 4096 4608
 .000 .000 .001 .002 .005 .000 .000 .000 .000 .000 .000

IP Flow Switching Cache, 4456448 bytes
1911 active, 63625 inactive, 131132988 added
1466401980 ager polls, 0 flow alloc failures
last clearing of statistics never
Protocol      Total Flows      Packets Bytes  Packets Active(Sec) Idle(Sec)
-----      Flows  /Sec      /Flow /Pkt      /Sec      /Flow      /Flow
TCP-Telnet    8399      0.0         28   64      0.1       12.5       15.7
TCP-FTP       13410     0.0         3    101     0.0        1.3        9.9
TCP-FTPD      780       0.0        173   704     0.0       20.0       3.9
TCP-WWW       829901    0.5         9   450     5.3        6.3        5.6
TCP-SMTP      170303    0.1         8   226     1.0        5.6        8.7
TCP-X         72        0.0        58   278     0.0       28.7       15.2
TCP-BGP       41217     0.0         5   281     0.1       13.5       15.9
TCP-NNTP      18        0.0         1    72     0.0        3.8       16.0
TCP-Frag      22003     0.0         1    32     0.0        1.3       16.0
TCP-other     2480106   1.7         6   139    10.8       3.1        4.2
UDP-DNS       98444883  68.1        3   100    230.3      4.8       16.0
UDP-NTP       136223    0.0         1    84     0.1        1.0       16.0
UDP-TFTP      45        0.0         2   155     0.0        2.0       16.0
UDP-Frag      43        0.0         1   494     0.0        0.1       15.9
UDP-other     28243323  19.5        2   161    51.3       4.2       16.0
ICMP          694753    0.4         6   138     3.3       10.0       16.0
IP-other      47897     0.0         6   115     0.2       17.5       15.9
Total:        131133376  90.7        3   119    303.0      4.7       15.7

SrcIf      SrcIPAddress      DstIf      DstIPAddress      Pr SrcP DstP  Pkts
Se2/0      192.169.33.3      Fa0/0      203.37.255.97     11 C4A8 0035 23
Fa0/0      203.37.255.97     Se2/0      192.169.33.3     11 0035 C4A8 23
Fa0/0      203.37.255.97     Se2/0      192.94.123.24    11 0035 9F05 1
Se2/0      192.94.123.24     Fa0/0      203.37.255.97     11 9F05 0035 1
Se2/0      148.184.176.31    Fa0/0      203.37.255.97     11 0035 0035 1
Fa0/0      203.37.255.97     Se2/0      148.184.176.31    11 0035 0035 1
Fa0/0      203.37.255.97     Se2/0      203.23.1.50       11 0035 ED76 1
Se2/0      203.23.1.50       Fa0/0      203.37.255.97     11 ED76 0035 1
Fa0/0      203.37.255.97     Se2/0      199.203.98.25     11 0035 82E5 2
Se2/0      199.203.98.25     Fa0/0      203.37.255.97     11 82E5 0035 2
Se2/0      202.96.237.81     Fa0/0      203.37.255.97     11 0035 0035 1
```

The first part of the output displays the packet size distribution of the traffic flows **into** the interfaces that NetFlow is configured on. The next portion of the output displays the flows, packet size, activity etc, for the flows per well known

Saturday, February 19, 2000

protocol. And the final section displays the source and destination interfaces/addresses/ports for the currently active traffic flows.

It is also possible to export this collected data to a system which will collect the data allowing the ISP to carry out further analysis. There is public domain software available (cflowd from Caida, NetFlowMet from the University of Auckland, for example), as well as commercial products such as Cisco's NetFlow Collector and Analyser packages.

To export the data, the following configuration commands are required:

```
ip flow-export version 5 [origin-as|peer-as]
ip flow-export destination x.x.x.x udp-port
```

The first command sets the export version to 5 (basically this includes BGP information such as AS number), and has options to include origin-as or peer-as in the exported records. The second command configures the IP address of the destination system, the NetFlow collector system, and the UDP port that the collector is listening on. Note that because the flow records use UDP it is important to design the infrastructure such that the flow collector is not too far away from the originating router. Some ISPs who use NetFlow for billing purposes build a separate management network simply to support this function.

A new feature as from 12.0(5)S is NetFlow aggregation where summarisation/aggregation of the Flow records is carried out on the router prior to the data being exported to the collecting system. The aim here is to reduce the amount of data going across the network from router to flow collector, thereby improving the reliability of the collecting system. Flow aggregation is enabled by the following commands:

```
ip flow-aggregation cache as|destination-prefix|prefix|protocol-port|source-prefix
enabled
export destination x.x.x.x UDP-port
```

Subcommands required include "enabled" which switches on the flow aggregation, and "export destination" which lists the host which will gather the aggregated records. The collector host needs to support NetFlow type 8 records to be able to read the aggregated information.

DNS and Routers

Mapping Domain Names to IP addresses is one of those commonly overlooked areas in a new ISP's operations. Doing a trace from Australia across the backbones in the US to a site in the UK gives you some thing like this:

```
traceroute to www.pipex.net (158.43.128.176): 1-30 hops, 38 byte packets
 1 brisbane-gw-fa20.cisco.com (144.254.153.1) 1.4 ms 0.911 ms 0.732 ms
 2 sydney-gb2.cisco.com (144.254.159.9) 29.2 ms 32.4 ms 35.9 ms
 3 sydney-wall-1.cisco.com (144.254.153.244) 35.8 ms 32.4 ms 36.4 ms
 4 telstra-gw.cisco.com (203.41.198.241) 35.2 ms 46.4 ms 32.8 ms
 5 Serial5-1-3.pad20.Sydney.telstra.net (139.130.34.205) 83.7 ms 37.2 ms 35.7 ms
 6 FastEthernet0-0-0.pad8.Sydney.telstra.net (139.130.249.228) 133 ms 43.9 ms 71.6 ms
 7 bordercore4-hssi0-0.SanFrancisco.cw.net (166.48.19.249) 260 ms 254 ms 276 ms
 8 core7.SanFrancisco.cw.net (204.70.4.93) 303 ms 260 ms 267 ms
 9 Hssi2-1-0.BR1.SCL1.Alter.Net (206.157.77.74) 284 ms 345 ms 433 ms
10 105.at-5-0-0.XR4.SCL1.ALTER.NET (152.63.48.182) 404 ms 496 ms 429 ms
11 195.at-1-0-0.TR4.SCL1.ALTER.NET (152.63.48.130) 324 ms 415 ms 308 ms
12 207.ATM6-0.TR2.NYC1.ALTER.NET (152.63.3.201) 343 ms 342 ms 335 ms
13 198.ATM6-0.XR2.NYC1.ALTER.NET (146.188.178.193) 333 ms 380 ms 359 ms
14 194.ATM3-0.GW1.NYC5.ALTER.NET (146.188.177.229) 347 ms 345 ms 333 ms
15 421.ATM4-0.BR1.NYC5.ALTER.NET (137.39.30.118) 384 ms 341 ms 337 ms
16 225.ATM6-0-0.CR1.LND2.Alter.Net (146.188.7.69) 407 ms 399 ms 409 ms
17 311.ATM2-0-0.GW2.LND2.Alter.Net (146.188.3.114) 423 ms 414 ms 418 ms
18 pos0-1.BR2.LND2.gbb.uu.net (146.188.5.50) 422 ms 415 ms 415 ms
19 srp4-0.cr1.uk2.london.pipex.net (158.43.233.1) 416 ms 409 ms 407 ms
20 pos0-2.cr1.doc.london.pipex.net (158.43.254.25) 408 ms 415 ms 422 ms
21 pos0-1.cr2.doc.london.pipex.net (158.43.254.66) 409 ms 414 ms 421 ms
22 pos4-0-0.cr2.cbgl.gbb.uk.uu.net (158.43.254.1) 405 ms 423 ms 408 ms
23 www.pipex.net (158.43.128.176) 408 ms * 414 ms
```

Notice that each of the router's IP addresses have a corresponding DNS entry. These very descriptive DNS names help Internet users and operators understand what is happening with their connections and which route the outbound traffic is taking. The descriptive names are an invaluable aid to troubleshooting problems on the Net.

Here are some examples of descriptive DNS formats used by various ISPs:

C&W	bordercore4-hssi0-0.SanFrancisco.cw.net
BBN Planet	p2-0.paloalto-nbr2.bbnplanet.net
Concert	core1-h1-0-0.uk1.concert.net
Sprint	sl-bb6-dc-1-1-0-T3.sprintlink.net
DIGEX	sjc4-core5-pos4-1.atlas.digex.net
Verio	p0-0-0.cr1.mtvwca.pacific.verio.net
IJ	otemachi5.ijj.net
Qwest	sfo-core-03.inet.qwest.net
Telstra BigPond	Pos5-0-0.cha-core2.Brisbane.telstra.net
UUNET	ATM2-0.BR1.NYC5.ALTER.NET
Teleglobe	if-8-0.core1.NewYork.Teleglobe.net
VSNL	E3-VSB1-LVSB.Bbone.vsnl.net.in

You can specify a default domain name that the Cisco IOS software will use to complete domain name requests. You can specify either a single domain name or a list of domain names. Any IP host name that does not contain a domain name will have the domain name you specify appended to it before being added to the host table.

```
ip domain-name name
ip domain-list name
```

It is also advisable to include a name server for the router to resolve DNS request:

```
ip name-server server-address1 [[server-address2]...server-address6]
```

Remember that the current practice on the Internet is to quote at least two DNS resolvers.

IOS and Loopback interfaces

There are many instances throughout this document where the use of the loopback interface is mentioned. While this is not a feature unique to IOS, there are many and considerable advantages in making full use of the capability the loopback interface allows. This section brings together all the occasions the loopback interface is mentioned in this document, and describes how they can be useful to the ISP Network Engineer.

Background

ISPs endeavour to minimise the unnecessary overhead present in their networks. This unnecessary overhead can be number of networks carried in the IGP, the number of skilled engineering staff to operate the network, or even network security. The utilisation of one feature, the loopback interface on the router, goes a long way to help with each of the three scenarios mentioned here.

The size of the IGP is attended to by summarisation of point to point addresses at PoP or regional boundaries, the use IP unnumbered on static WAN interfaces, and a carefully designed network addressing plan. ISP Network security is of paramount importance, and any techniques which make the management simpler are usually welcomed. For example, when routers access core servers for whatever reasons, ISPs apply filters or access-lists to these servers so that the risk of compromise from the outside is reduced. The loopback interface is helpful here too.

Saturday, February 19, 2000

It is very common to assign all the IP addresses used for loopback interfaces from one address block. For example, an ISP with around 200 routers in their network may assign a /24 network (253 usable addresses) for addressing the loopback interface on each router. If this is done, all dependent systems can be configured to permit this address range access the particular function concerned, be it security, unnumbered WAN links, or the iBGP mesh. Some examples follow in the rest of this section.

BGP Update-Source

Here the iBGP mesh is built using the loopback interface on each router. The loopback doesn't disappear, ever, so results in a stable iBGP, even if the underlying physical connectivity is less than reliable. Sample configuration:

```
hostname gateway1
!
interface loopback 0
 ip address 215.17.1.34 255.255.255.255
!
router bgp 200
 neighbor 215.17.1.35 remote-as 200
 neighbor 215.17.1.35 update-source loopback 0
 neighbor 215.17.1.36 remote-as 200
 neighbor 215.17.1.36 update-source loopback 0
!
```

Router ID

If a loopback interface is configured on the router, it's IP address is used as the router ID. This is important to ensure stability and predictability in the operation of the ISP's network.

OSPF chooses the Designated Router on a LAN as the device which has the highest IP address. If routers are added or removed from the LAN, or a router gains an interface with a higher address than that of the existing DR, it is likely that the DR will change during the next network event – this may or may not be undesirable. It can be avoided by ensuring that the loopback interface is configured and in use on all routers on the LAN.

The loopback interface is used for the BGP router ID. If the loopback isn't configured, then BGP uses the lowest IP address on the router. Again due to the ever-changing nature of an ISP network, this value may change, possibly resulting in operational confusion. Configuring and using a loopback interface ensures stability.

Important Note: If the router has two or more loopback interfaces configured, the router ID is the highest IP address of the configured loopback interfaces.

IP Unnumbered Interfaces

IP addresses should not be used on static WAN links to customers. IP unnumbered saves /30 of address space, and one entry in the IGP routing table, a significant saving for a large number of customers. IP unnumbered makes use of the loopback interface on the ISP's backbone router, the same loopback interface used for iBGP etc. An example configuration:

ISP's router:

```
interface loopback 0
 description Loopback interface on Gateway Router 2
 ip address 215.17.3.1 255.255.255.255
!
interface Serial 5/0
 description 128K HDLC link to Galaxy Publications Ltd [galpub1] WT50314E R5-0
 bandwidth 128
 ip unnumbered loopback 0
!
```



```
ip route 215.34.10.0 255.255.252.0 Serial 5/0
!
```

Customer's router:

```
interface Ethernet 0
  description Galaxy Publications LAN
  ip address 215.34.10.1 255.255.252.0
!
interface Serial 0
  description 128K HDLC link to Galaxy Internet Inc WT50314E C0
  bandwidth 128
  ip unnumbered ethernet 0
!
ip route 0.0.0.0 0.0.0.0 Serial 0
!
```

Exception Dumps by FTP

Cisco routers can be configured to dump core memory to an FTP server as part of the diagnostic and debugging process. However, this core dump should be to a system not running a public ftp server, but one heavily protected by filters (tcp wrapper even) which only allows the routers access. If the loopback interface address is used as source address from the router, and is part of one address block, the filter is very easy to configure. A 200 router network with 200 disparate IP addresses makes for a very large filter list on the ftp server. Sample IOS configuration:

```
ip ftp source-interface Loopback0
ip ftp username cisco
ip ftp password 7 045802150C2E
exception protocol ftp
exception dump 169.223.32.1
```

TFTP-SERVER Access

TFTP is the most common tool to upload and download configurations. The TFTP Server's security is critical. That means using security tools with a IP source addresses. IOS allows tftp to be configured to use a specific IP interfaces address. This allows a fixed ACL on the TFTP Server based on a fixed address on the router (i.e. the loopback interface).

```
ip tftp source-interface Loopback0
```

SNMP-SERVER Access

If SNMP is used in the network, the loopback interface can again be brought into use for security access issues. If SNMP traffic from the router is sourced from its loopback interface, it is easy to protect the SNMP management station in the NOC. Sample IOS configuration:

```
access-list 98 permit 215.17.34.1
access-list 98 permit 215.17.1.1
access-list 98 deny any
snmp-server community 5nmc02m RO 98
snmp-server trap-source Loopback0
snmp-server trap-authentication
snmp-server host 215.17.34.1 5nmc02m
snmp-server host 215.17.1.1 5nmc02m
```

TACACS/RADIUS-Server Source Interface

Most ISP uses TACACS+ or Radius for user authentication. Very few define accounts on the router itself as this offers more opportunity for the system to be compromised. A well-protected TACACS+ server accessed only from the router's loopback interface address block offers more security of user and enable accounts. Sample configuration for standard and enable passwords:

Saturday, February 19, 2000

```
aaa new-model
aaa authentication login default tacacs+ enable
aaa authentication enable default tacacs+ enable
aaa accounting exec start-stop tacacs+

ip tacacs source-interface Loopback0
tacacs-server host 215.17.1.2
tacacs-server host 215.17.34.10
tacacs-server key CKr3t#
```

When using RADIUS, either for user administrative access to the router, or for dial user authentication and accounting, the router configuration to support loopback interfaces as the source address for RADIUS packets originating from the router looks like this:

```
radius-server host 215.17.1.2 auth-port 1645 acct-port 1646
radius-server host 215.17.34.10 auth-port 1645 acct-port 1646
ip radius source-interface Loopback0
```

NetFlow Flow-Export

Exporting traffic flows from the router to a NetFlow Collector for traffic analysis or billing purposes is quite a common activity nowadays. Using the loopback interface as the source address for all exported traffic flows from the router allows for more precise and less costly filtering at or near the server. A configuration example:

```
ip flow-export destination 215.17.13.1 9996
ip flow-export source Loopback0
ip flow-export version 5 origin-as

interface Fddi0/0/0
description FDDI link to IXP
ip address 215.18.1.10 255.255.255.0
ip route-cache flow
ip route-cache distributed
no keepalive
!
```

Here interface FDDI0/0/0 has been configured to capture flow records. The router has been configured to export version 5 style flow records to the host at IP address 215.17.13.1 on UDP port 9996, with source address being the router's loopback interface.

NTP Source Interface

NTP is the means of keeping the clocks on all the routers on the networked synchronised to within a few milliseconds. If the loopback interface is used as the source interface between NTP speakers, it make filtering and authentication somewhat easier to maintain. Most ISPs only wish to permit their customers to synchronise with their time servers, not everyone else in the world. A configuration example:

```
clock timezone SST 8
!
access-list 5 permit 192.36.143.150
access-list 5 permit 169.223.50.14
!
ntp authentication-key 1234 md5 104D000A0618 7
ntp authenticate
ntp trusted-key 1234
ntp source Loopback0
ntp access-group peer 5
ntp update-calendar
ntp peer 192.36.143.150
ntp peer 169.223.50.14
!
```

SYSLOG Source Interface

Syslog servers also require careful protection on ISP backbones. Most ISPs prefer to see only their own systems' syslog messages, not anything from the outside world. And denial of service attacks on syslog devices are not unknown either. Protecting the syslog server is again made easier if the known source of syslog messages comes from a well-defined set of address space, for example that used by the loopback interfaces on the routers. A configuration example:

```
logging buffered 16384
logging trap debugging
logging source-interface Loopback0
logging facility local7
logging 169.223.32.1
```

Telnet to the Router

This may seem to be an odd example in a document dedicated to IOS Essentials. However, remember that a loopback interface on a router never changes its state, and has a rare if no need to change IP address. Physical interfaces may be physically swapped out or renumbered, address ranges may change, but the loopback interface will always be there. So if the DNS is set up so that the router name maps to the loopback interface address, there is one less change to worry about during operational and configuration changes elsewhere in the ISP backbone. And ISP backbones are continuously developing entities. Example from the DNS forward and reverse zone files:

```
; net.galaxy zone file
net.galaxy.      IN      SOA      ns.net.galaxy. hostmaster.net.galaxy. (
                1998072901 ; version == date(YYYYMMDD)+serial
                10800      ; Refresh (3 hours)
                900        ; Retry (15 minutes)
                172800     ; Expire (48 hours)
                43200      ; Mimimum (12 hours)
                )
                IN      NS      ns0.net.galaxy.
                IN      NS      ns1.net.galaxy.
                IN      MX      10 mail0.net.galaxy.
                IN      MX      20 mail1.net.galaxy.
;
localhost      IN      A        127.0.0.1
gateway1       IN      A        215.17.1.1
gateway2       IN      A        215.17.1.2
gateway3       IN      A        215.17.1.3
;
;etc etc
; 1.17.215.in-addr.arpa zone file
;
1.17.215.in-addr.arpa. IN      SOA      ns.net.galaxy. hostmaster.net.galaxy. (
                1998072901 ; version == date(YYYYMMDD)+serial
                10800      ; Refresh (3 hours)
                900        ; Retry (15 minutes)
                172800     ; Expire (48 hours)
                43200      ; Mimimum (12 hours)
                )
                IN      NS      ns0.net.galaxy.
                IN      NS      ns1.net.galaxy.
1                IN      PTR      gateway1.net.galaxy.
2                IN      PTR      gateway2.net.galaxy.
3                IN      PTR      gateway3.net.galaxy.
;
;etc etc
```

On the router, set the telnet source to the loopback interface:

```
ip telnet source-interface Loopback0
```

Saturday, February 19, 2000

RCMD to the router

RCMD requires the operator to have the Unix rlogin/rsh clients to allow access to the router. Some ISPs use RCMD for grabbing interface statistics, uploading or downloading router configurations, or the taking a snapshot of the routing table. The router can be configured so that RCMD connections use the loopback interface as the source address of all packets leaving the router:

```
ip rcmd source-interface Loopback0
```

SECURITY

This section on IOS Security Features assumes that the ISP Engineer has a working grasp of the fundamentals of system security. If not, please review the materials listed below to help gain an understanding of some of the fundamentals. Also, the sections below are intended to supplement the Cisco Documentation. It is assumed that the ISP Engineer will Read The Fantastic Manuals (RTFM ☺) in parallel with this whitepaper.

Increasing Security on IP Networks. An old, but very important document on some of the security essentials in IP based networks.

<http://www.cisco.com/univercd/cc/td/doc/cisintwk/ics/cs003.htm>

Cisco's INTERNET SECURITY ADVISORIES. An online list of all of Cisco's Security advisories. It includes tutorials and details on how to protect yourself from some of the ugliness on the Internet today.

<http://www.cisco.com/warp/customer/707/advisory.html>

Cisco's IOS Documentation – 11.2 Security Configuration Guide. The documentation with the 11.2 software release reorganised many of the security features of IOS into their own chapter. This documentation is the foundation for 12.0. Available on the *Cisco Documentation CD* or publicly on-line via Cisco Connection On-Line (CCO):

http://www.cisco.com/univercd/data/doc/software/11_2/2cbook.htm

RFC 2196 Site Security Handbook. B. Fraser. September 1997. (Format: TXT=191772 bytes) (Obsoletes RFC1244) (Also FYI0008) (Status: INFORMATIONAL) One of the most useful starting places for Internet security.

<http://info.internet.isi.edu:80/in-notes/rfc/files/rfc2196.txt>

Craig Huegen's SMURF page. A very useful resource for ISPs to learn how to protect themselves from the many flavours of denial of service attacks.

<http://www.quadrunner.com/~chuegen/smurf.txt>

Jared Mauch's [jared@puck.nether.net] Smurf Sweep Results Page. Jared has scanned large sections of the Internet looking for networks that could be used as smurf amplifiers. This page lists his results and provides a way to check IP prefixes and AS numbers.

<http://puck.nether.net/smurf-check/>

Security for an ISP

Securing an enterprise network from Internet threats is easy when compared with the problems of security facing an ISP. When an enterprise network connects to the Internet, there is essentially one Internet security problem – protecting your network from outside intrusion. To achieve their Internet security objectives, an enterprise will balance tradeoffs with connectivity, accessibility, performance, and security.

An ISP's security concerns are much broader. The ISP business is all about transparent, cost effective and high performance Internet connectivity. Security measures will affect the ISP's network. Yet, at the same time, security threats are real. ISPs are very visible targets for malicious, vindictive, and criminal attacks. ISPs must protect themselves, help protect their customers, and minimise the risk of their customers from becoming problems to others on the Internet.

This section describes tools that all ISPs should consider for their overall security architecture. Most of these tools are *passive* tools. Once configured, they will help prevent security problems from happening and make it more difficult to cause mischief on the ISP's network.

Saturday, February 19, 2000

Global Services that are not needed or are a security risk

Many of the built in services in IOS are not needed in an ISP backbone environment. These features should be turned off in your default configuration. Only turned them on if there are explicit requirements.

```
no service finger
no service pad
no service udp-small-servers
no service tcp-small-servers
no ip bootp server
```

Some of these will be pre-configured in IOS (depending on release) to be turned off by default, but ISPs should ensure they are explicitly turned off in the master configuration files.

The whitepaper/field alert *Defining Strategies to Protect Against UDP Diagnostic Port Denial of Service Attacks* describes the security risk and provides pointers to public discussion on the ISP Operations forums. This whitepaper is posted publicly at: <http://www.cisco.com/warp/public/707/3.html>.

no service finger disables the process which listens for “finger” requests from remote hosts. Only ISP personnel normally access the backbone routers, and there are other and better means of tracking who is logged in. Besides, “finger” is a known security risk in the Internet, due to its divulgence of detailed information of people logged into a system. (In 12.0 and later releases, this command has been changed to “no ip finger”.)

no service pad is simply not required. It refers back to the days of X.25 networking, and in recent versions of IOS has become the default.

The small TCP and UDP servers are those with port numbers below 10 – typical services include “echo” and “discard” ports, the former echoing all packets sent to it, the latter throwing away all packets sent to it. If they are enabled and active, they could be used to carry out successful denial of service attacks – their use will divert CPU resources away from other processes which will cause problems for the connected networks and Internet service dependent on that router.

The *bootp* service provides support for systems which find their configuration using the bootp process. This is commonly used in LANs (X-terminals commonly use bootp for example), and never on the WAN. It should be disabled.

Interface Services that are not needed or are a security risk

Some IP features are great for Campus LANs, but do not make sense on an ISP backbone. Abuse of these functions by “cyberpunks” increases the ISP’s security risk. All interfaces on an ISP’s backbone router should have the following configured by *default*:

```
no ip redirects
no ip directed-broadcast
no ip proxy-arp
```

The configuration “no ip redirects” means that the router will not send redirect messages if the IOS is forced to resend a packet through the same interface on which it was received.

The configuration command “no ip directed-broadcast” means that the translation of directed broadcast to physical broadcasts is disabled. If enabled, a broadcast to a particular network could be directed at a router interface, producing effects which may be undesirable and potentially harmful. An example of the ill effects of directed broadcasts being enabled is the so-called SMURF attack. For more information about SMURF, see Craig Heugen’s SMURF website at <http://www.quadrunner.com/~chuegen/smurf.txt>. As from IOS 12.0, “no ip directed-broadcast” has become the default on all router interfaces.

The configuration “no ip proxy-arp” means that the router will not respond to ARP requests for other hosts on the network connected to this interface if it knows that MAC address of those hosts. This again is to prevent undesirable effects on the connected network, and potential security problems.

Cisco Discovery Protocol

The Cisco Discovery Protocol (CDP) is used for some network management functions on campus LANs etc. It basically allows any system on a directly connected segment to discover that the equipment is manufactured by Cisco (could be a router, switch, etc), and determine information such as the model number and the software version running. This is very useful in some instances but does not make very much sense on an ISP’s backbone where the ISP should be completely aware of what is installed and what software versions are running!

The information available from CDP does not threaten security as such, but information such as software version could be used by attackers to exploit known bugs and harm the operation of the ISP’s network.

CDP may be disabled using the global command:

```
no cdp run
```

If CDP is required on an ISP’s network, for what ever reason, it is possible to leave CDP running, but disable the protocol on a per interface basis. The interface configuration command:

```
no cdp enable
```

will disable CDP on a particular interface. It is strongly recommended that CDP be disabled on all public-facing interfaces, whether those face exchange points, upstream ISP, or even customers.

Note that CDP is enabled by default on 11.1CA, 11.1CC, and more recent software.

Login Banners

Much overlooked, but important in the age of the commercial ISP is the *banner login* command. This feature is part of the “banner” command set, which displays text when users connect to the router. *Banner login* displays text when a user first initiates a telnet session to the router.

It may be seemingly trivial, but a lack of banner is as effective as a security device as a banner telling connected sessions that only those who are authorised to are permitted to connect. Some ISPs are now using banners with message content similar to the one below. Any ISP should consider whether their interest is served best by including a banner with an official warning, or nothing at all. It is good practice not to identify too much about the system itself in the banner. (Things like **joes-router** may not be such a good idea as they may give a hint about the user/owner of the system, and any user-ids or passwords on it.)

```
banner login ^
  Authorised access only
  This system is the property of Galactic Internet
  Disconnect IMMEDIATELY if you are not an authorised user!
  Contact noc@net.galaxy +99 876 543210 for help.
^
```

Saturday, February 19, 2000

Another type of banner available is the “exec” banner, displayed at the time a user has successfully authenticated and logged in. For example, a note to all engineering staff on a backbone router:

```
banner exec ^
PLEASE NOTE - THIS ROUTER SHOULD NOT HAVE A DEFAULT ROUTE!
It is used to connect paying peers. These `customers' should not be able to default to us.
The configuration of this router is NON-STANDARD
Contact Network Engineering +99 876 543234 for more information.
^
```

Use *enable secret*

Use *enable secret* in lieu of the *enable password* command. The encryption algorithm type 7 used in *enable password* and *service password-encryption* is reversible. The *enable secret* command provides better security by storing the enable secret password using a non-reversible cryptographic function. The added layer of security encryption it provides is useful in environments where the password crosses the network or is stored on a TFTP server.

```
service password-encryption
enable secret <removed>
no enable password7
```

Almost all passwords and other authentication strings in Cisco IOS configuration files are encrypted using the weak, reversible scheme used for user passwords. To determine which scheme has been used to encrypt a specific password, check the digit preceding the encrypted string in the configuration file. If that digit is a 7, the password has been encrypted using the weak algorithm. If the digit is a 5, the password has been hashed using the stronger MD5 algorithm. Even though *enable secret* is used for the enable password, do not forget *service password-encryption* so that the remaining passwords are stored in the configuration with type 7 encryption rather than in plain text. Weak encryption is better than none at all.

For example, in the configuration command:

```
enable secret 5 $l$iUjJ$cDZ03KKGh7mHfX2RSbDqP.
```

The *enable secret* has been hashed with MD5, whereas in the command:

```
username jbash password 7 07362E590E1B1C041B1E124C0A2F2E206832752E1A01134D
```

the password has been encrypted using the weak reversible algorithm. Since there are several versions of code designed to break the weak encryption on a Cisco, ISPs are strongly encouraged to use other strategies for passwords that are not protected by strong encryption. Cisco IOS supports Kerberos, TACACS+, and RADIUS authentication architectures, so the option is open to use AAA to get into the router versus having usernames on the router itself.

A Cisco Technical Tip, *Cisco IOS Password Encryption Facts*, explains the security model behind Cisco password encryption, and the security limitations of that encryption. It is publicly on-line via Cisco Connection On-Line (CCO):

<http://www.cisco.com/warp/public/701/64.html>

⁷ A caveat though. Do not remove the *enable password* as above if the boot ROMs or boot image of the router does not support the *enable secret* configuration. The use of secret is supported in IOS 11.0 and later. With an older boot ROM and no *enable password* it is possible to gain access to the router without supplying any password should the router end up running the boot image due to some network problem, or malfunction. A network's first line of defense are the routers used, and anyone wishing to compromise a network will more than likely start with the router rather than any system behind that router (where configurations might be stored).

Turn on Nagle

The Nagle congestion control algorithm is something that many ISPs turn on to improve the performance of their telnet session to and from the router. When using a standard TCP implementation to send keystrokes between machines, TCP tends to send one packet for each keystroke typed. On larger networks, many small packets use up bandwidth and contribute to congestion.

John Nagle's algorithm (RFC 896) helps alleviate the small-packet problem in TCP. In general, it works this way: The first character typed after connection establishment is sent in a single packet, but TCP holds any additional characters typed until the receiver acknowledges the previous packet. Then the second, larger packet is sent and additional typed characters are saved until the acknowledgement comes back. The effect is to accumulate characters into larger chunks, and pace them out to the network at a rate matching the round-trip time of the given connection. This method is usually a good for all TCP-based traffic, and helps when connectivity to the router is poor or congested, or the router itself is busier than normal. However, do not use the *service nagle* command if you have XRemote users on X Window sessions.

```
service nagle
```

The Ident Feature

Identification (*ident*) support allows you to query a Transmission Control Protocol (TCP) port for identification. This feature enables an insecure protocol, described in RFC 1413, to report the identity of a client initiating a TCP connection and a host responding to the connection. No attempt is made to protect against unauthorised queries. This command should only be enabled if the consequences and the advantages in the local situation are understood.

```
ip ident
```

Some ISP Backbone engineers like IDENT. Others do not. New ISP Engineers are recommended to look into IDENT, read the RFC, try it, and see if it fits as a security tool on your backbone.

System Access

Principles

Access to routers to carry out administrative functions can be achieved either physically through the console port, or remotely through a VTY (virtual terminal). The console port is generally only used as last resort access, with the common set-up being that the console is plugged into a "console server" which gives the ISP what is called out of band access. Most versions of IOS have support for 5 VTY ports on the router – these are the most common way of accessing the device, operating across the network and supporting multiple protocols. ISP's commonly use telnet, with support for SSH (Secure Shell) support added from the 12.0S software release.

The following configuration guidelines are common sense:

1. Use Access Control Lists (ACLs) to restrict telnet attempts to be from source networks you trust. This is not foolproof, but it adds a layer of difficulty. It is also recommended that you include *anti-spoofing* filters on the edge of your network to prevent spoof attempts from outside your network.
2. Implement *username/password* pairs instead of the traditional password only technique of logging into a router. Using both a username and password increases the level of effort need to use brute force to crack the password. Ideally, an AAA protocol (Radius, TACACS+, or Kerberos) should be used. If an AAA protocol is used, the *username/password* pair could be used as a backup in case the AAA protocol is not working.

Saturday, February 19, 2000

3. Include shorter inactivity timeouts. The inactivity timeout minimises some of the risk when the careless operator leaves their terminal logged into the router.

Each of these will now be looked at in more detail in the following sections.

VTY and Console Port Timeouts

By default the timeout applied to all connections to the VTY, Console and AUX ports on a router is 10 minutes. This timeout is controlled by the `exec-timeout` command, as in this example:

```
line con 0
  exec-timeout 5 0
line aux 0
  exec-timeout 10 0
line vty 0 4
  exec-timeout 5 0
```

Here the router has been told to disconnect console port and vty connections which have been idle for more than 5 minutes and 0 seconds. The auxiliary port timeout has been set to 10 minutes.

Notice that setting the idle timeout to 0 means that the session will be left connected indefinitely. This is generally regarded as bad practice as this will hog the few available ports on the router, and could cause maintenance access problems in the event of emergencies.

Furthermore, enabling TCP keepalives on incoming connections will ensure that any sessions left hanging by a remote system crash or disconnection won't block or use up the available router VTY ports. The configuration command:

```
service tcp-keepalives-in
```

ensures this.

Access List on the VTY Ports

It is important to secure the vty ports used for telnet access with a standard ACL. By default there are no access controls on any of the VTY ports. If left this way and a password is applied⁸ to the VTY port the router would be wide open to all comers to attempt a brute force crack against the password. The configuration below with *access-list 3* is typical:

```
aaa new-model
aaa authentication login Cisco-Lab local
!
username Cisco1 password 7 11041811051B13
!
access-list 3 permit 215.17.1.0 0.0.0.255
access-list 3 permit 215.17.34.0 0.0.0.255
access-list 3 deny any
!
line vty 0 4
  access-class 3 in
  exec-timeout 5 0
  transport input telnet ssh
  transport output none
  transport preferred none
  login authentication Cisco-Lab
  history size 256
!
```

Access-list 3 defines a network 215.17.1/24 and 215.17.34/24 as the only networks with access to these vtys (these networks could be the administration or NOC networks at two locations, for example).

⁸ Telnet is forbidden to any VTY port that does not have a password or some other authentication configured.

A *timeout* of 5 minutes is applied to the interface. The second field is for “seconds”, for finer granularity. ISPs generally pick the best timeout values according to experience and the operating environment. See the next section for more sophisticated means of protecting access.

Also, all unnecessary *transports* are removed – users of vtys only require character access to the router, nothing else. (Other available transports such as pad, rlogin, and V120, not required on an ISP backbone router.) It is good practice to configure necessary transports on a per interface application basis – dialup users will only require IP transport, for example. In the case above, only telnet has been permitted to the vty port, and no outbound connections are permitted.

If the router supports more than 5 vtys, don’t forget them! IP only software (-i- and -p- code releases) only support 5, but other feature sets can support 64 or as many as 1024 vtys. Be sure to apply access lists to all of them if they are configured. The command `line vty 0 4` will cover the first five vty ports.

VTY Access and SSH

Prior to 12.0S software the only method really used to access the VTY ports was telnet. Rlogin has been used by some ISPs, especially for executing one-off commands but the protocol is insecure and can’t be recommended. SSH version 1 support has now been added, giving ISPs greater flexibility and some security when accessing their equipment across the Internet.

Before SSH can be configured, the router needs to be running a cryptographic image which supports SSH. The standard Service Provider (-p-) images do not have SSH support as the export of DES and 3DES is restricted by the US Government. The cryptographic images are made available on CCO after an approval application has been submitted. The approval application form can be found at http://www.cisco.com/cgi-bin/crypto/crypto_main.pl. Once the application has been approved, permission will be granted to download the necessary images.

Once the appropriate cryptographic image is running, SSH needs to be set up on the router. The following sequence of configuration commands gives an example of how this may be achieved:

```
beta7200(config)#crypto key generate rsa
```

and select a key size of at least 1024 bits. After this, add `ssh` as the input transport on the vtys:

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

It is now possible to use SSH to access the router. If the IOS image supporting DES is being used, the SSH client on the operator’s system needs to support DES (most SSH clients have this disabled by default as it is no longer considered very secure). This may require the client to be recompiled (if this is an option).

Note: a username/password pair **must** be configured on the router before SSH access will work. However, it is strongly recommended that AAA is used to authenticate users (see below) as this is the preferred way of securing the router.

```
ssh beta7200 -l philip
```

is the Unix command to connect to the router `beta7200` using `ssh` with the username “philip”.

User authentication

It is good practice to register each individual user with a separate user-id on each router. If a generic account is set up, it is easier for it to fall into the wrong hands, and there is virtually no accountability, resulting in abuse of access, and potential malfunction of the network. In addition, if the default password only login is used, it becomes very easy to use a brute force crack utility to get the password. A *username/password* pair makes brute force techniques harder, but not impossible.

This example shows one way of configuring user-ids. It is practical for networks of a few routers, but does not scale, and suffers from the weak type-7 encryption. (It’s therefore best avoided, but given here for completeness.)

Saturday, February 19, 2000

Configuring:

```
username joe password 7 045802150C2E
username jim password 7 0317B21895FE
!
line vty 0 4
  login local
!
```

on the router will change the login prompt sequence from:

```
Password:
to:
Username:
Password:
```

where the username requested is from those listed above. Each user will have to supply the password on request.

Using AAA to Secure the Router

The preferred and recommended method is to use an AAA protocol like TACACS+, Radius, or Kerberos. Here all the users who have access to the routers have their usernames and passwords held at a central location, off the router. This has several advantages:

- Recall that the encryption method 7 is reversible. Anyone who has access to the router configuration could potentially work out the password and gain access to the system.
- Scalability. If there is a new user, or a user leaves the ISP, it is easy to change the password database once. Changing it on many different routers becomes a considerable task.
- Passwords are held in Unix encrypted format on the central AAA server. The algorithm for Unix password encryption is not reversible, so is more secure.
- All accesses are logged to the AAA server. In fact, some AAA software will allow all actions on the router to be logged.

A server for Windows is available at <http://www.hkstar.com/~unet/> – note that this one is not a Cisco product. Also, there is the commercial Cisco ACS software available for Windows NT and Sun Solaris systems. A TACACS+ server for Unix platforms is available on Cisco's Engineering ftp site at ftp://ftp-eng.cisco.com/pub/tacacs/tac_plus.F4.0.3.alpha.tar.Z. For more information about AAA, TACACS+ and user authentication, check the CCO web pages, for example, at http://www.cisco.com/warp/customer/728/Secure/cseac_ds.htm.

On the router, a typical tacacs+ configuration would be:

```
aaa new-model
aaa authentication login default tacacs+ enable
aaa authentication enable default tacacs+ enable
aaa accounting exec start-stop tacacs+

ip tacacs source-interface Loopback0
tacacs-server host 215.17.1.2
tacacs-server host 215.17.34.10
tacacs-server key CKr3t#
```

To explain these commands, the `authentication login` states that TACACS+ should be used for login authentication – if the TACACS+ servers are not reachable, the local `enable SECRET` is used. The `authentication enable` command states that TACACS+ should be used to authenticate the use of the `ENABLE` command. The `enable password` should be taken from the TACACS+ server before using the local `enable SECRET`.

Note the use of the Loopback interface as the source of TACACS+ requests (reasons as in previous examples), and the use of two TACACS+ servers for redundancy and resilience.

If the router is running an IOS version which does not support TACACS+ (pre 11.0), it is strongly advised that it is upgraded to at least 11.0, as the more recent software has more features appropriate to ISP's as discussed in this document.

Router Command Auditing

AAA accounting on the router and TACACS+ server can be configured to track all commands or a limited set of commands typed into the router. AAA Command accounting provides information about the EXEC shell commands for a specified privilege level that are being executed on a router. Each command accounting record includes a list of the commands executed for that privilege level, as well as the date and time each command was executed, and the user who executed it.

The following example shows the information contained in a TACACS+ command accounting record for privilege level 1:

```
Wed Jun 25 03:46:47 1997      172.16.25.15    fgeorge  tty3    5622329430/4327528  stop
task_id=3      service=shell  priv-lvl=1    cmd=show version <cr>
Wed Jun 25 03:46:58 1997      172.16.25.15    fgeorge  tty3    5622329430/4327528  stop
task_id=4      service=shell  priv-lvl=1    cmd=show interfaces Ethernet 0 <cr>
Wed Jun 25 03:47:03 1997      172.16.25.15    fgeorge  tty3    5622329430/4327528  stop
task_id=5      service=shell  priv-lvl=1    cmd=show ip route <cr>
```

The following example shows the information contained in a TACACS+ command accounting record for privilege level 15:

```
Wed Jun 25 03:47:17 1997      172.16.25.15    fgeorge  tty3    5622329430/4327528  stop
task_id=6      service=shell  priv-lvl=15   cmd=configure terminal <cr>
Wed Jun 25 03:47:21 1997      172.16.25.15    fgeorge  tty3    5622329430/4327528  stop
task_id=7      service=shell  priv-lvl=15   cmd=interface Serial 0 <cr>
Wed Jun 25 03:47:29 1997      172.16.25.15    fgeorge  tty3    5622329430/4327528  stop
task_id=8      service=shell  priv-lvl=15   cmd=ip address 1.1.1.1 255.255.255.0 <cr>
```

Configuration control and audit of who is done what when on the routers is the key objective for using AAA command accounting on an ISP's backbone.

```
aaa new-model
aaa authentication login default tacacs+ enable
aaa authentication enable default tacacs+ enable
aaa accounting command 15 start-stop tacacs+
aaa accounting exec start-stop tacacs+

ip tacacs source-interface Loopback0
tacacs-server host 215.17.1.2
tacacs-server host 215.17.34.10
tacacs-server key CKr3t#
```

Full Example

The following is a full example putting all the techniques together.

```
service password-encryption
!
aaa new-model
aaa authentication login default tacacs+ enable
aaa authentication login Cisco-Lab local enable
aaa authentication enable default tacacs+ enable
aaa accounting exec start-stop tacacs+
!
username Cisco1 password 7 11041811051B13
```

Saturday, February 19, 2000

```
enable secret <removed>
!
access-list 3 permit 215.17.1.0 0.0.0.255
access-list 3 permit 215.17.34.0 0.0.0.255
access-list 3 deny any
!
ip tacacs source-interface Loopback0
tacacs-server host 215.17.1.2
tacacs-server host 215.17.34.10
tacacs-server key CKr3t#
!
line vty 0 4
access-class 3 in
exec-timeout 5 0
transport input telnet ssh
transport output none
transport preferred none
login authentication Cisco-Lab
history size 256
!
```

Egress and Ingress Filtering

Egress and ingress filtering are a critical part of an ISP's router configuration strategy. Ingress filtering applies filters to traffic coming into a network from outside (see Figure 2). This can be from an ISP's customers and/or from the Internet at large. Egress Filtering applies a filter for all traffic leaving an ISP's network (see Figure 2).

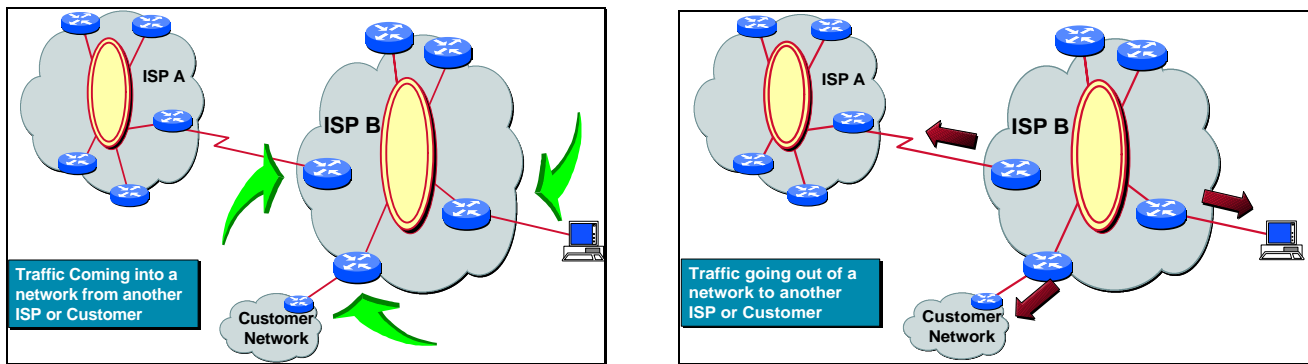


Figure 2 – Ingress and Egress Filtering

Both filtering techniques help protect an ISP's resources, its customers' networks, allows it to enforce policy, and minimises the risk of being the network chosen by hackers to launch an attack on other networks. ISPs are strongly encouraged to develop strategies using egress and ingress filtering to protect themselves from their customers and the Internet at large. By protecting themselves, the ISPs are working towards protecting the Internet in general.

RFC2267 *Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing*: P. Ferguson, D. Senie, January 1998, <http://info.internet.isi.edu/in-notes/rfc/files/rfc2267.txt> provides general guidelines for all ISPs on Ingress and Egress filtering.

There are several types of egress and ingress filtering – routing, packet, and dial-up access.

Ingress and Egress Route Filtering

Not all networks are supposed to be advertised on the Internet. RFC 1918 (Private address space), host subnet (127.0.0.0/8), and multicast addresses, for now, should be filtered from your advertisement to and from the Internet⁹. When you filter routes going to the Internet, it is called *egress filtering*. When you filter routes coming from the Internet, it is called *ingress filtering*. It is recommended that ISPs do both on their border routers. Here is an example configuration and filter:

```
router bgp 200
  no synchronisation
  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
!
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 192.168.0.0 0.0.255.255 255.255.0.0 0.0.255.255
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 224.0.0.0 31.255.255.255 224.0.0.0 31.255.255.255
access-list 150 permit ip any any
```

Ingress and Egress Packet Filtering¹⁰

Denial of service, spoofing, and other forms of attacks are on the increase on the Internet. Many of these attacks can be thwarted through the judicious use of ingress (packet originating from your network) and egress (packets arriving from the Internet) filtering.

NOTE: There could be a performance impact on the forwarding speed of the router when many filters are applied. Cisco's newer switching technologies help minimise the performance impact. For example, NetFlow switching is extremely efficient minimising the performance impact of very long IP access list. Due care and consideration should be taken whenever the access list starts getting beyond 50 entries. However, it should be stated that trading a few microseconds of IP forwarding speed for the safety of minimising the impact of denial of service attacks may prove to be worth while.

Egress Filtering – Preventing Transmission of Invalid IP Addresses

By filtering packets on your routers that connect your network to the Internet (Figure 3), you can permit only packets with valid source IP addresses to leave your network and get into the Internet. For example, if your network consists of network 165.21.0.0, and your router connects to your ISP using a serial 0/1 interface, you can apply the access-list as follows:

```
access-list 110 permit ip 165.21.0.0 0.0.255.255 any
access-list 110 deny ip any any log

interface serial 0/1
```

⁹ Experiments with Multicast BGP (MBGP) have now begun on some ISP sites. Deployment of MBGP will require a re-think and re-design of the egress/ingress route filters.

¹⁰ CAVEAT: This section covers the simple case single homed downstream customers. More thought is required when applying packet filtering for ISPs who are multihomed. For example, when the customer ISP's link to your network goes down, you will see packets from that network coming from "the Internet". Also be aware that a multihomed customer may require special routing to implement loadsharing – in that case you will again see that ISP's traffic coming from "the Internet".

Saturday, February 19, 2000

```
ip access-group 110 out
```

The last line of the access-list determines if there is any traffic with an invalid source address entering the Internet. If there are any matches, they will be logged. It is not crucial to have this line, but it will help locate the source and extent of the possible attacks.

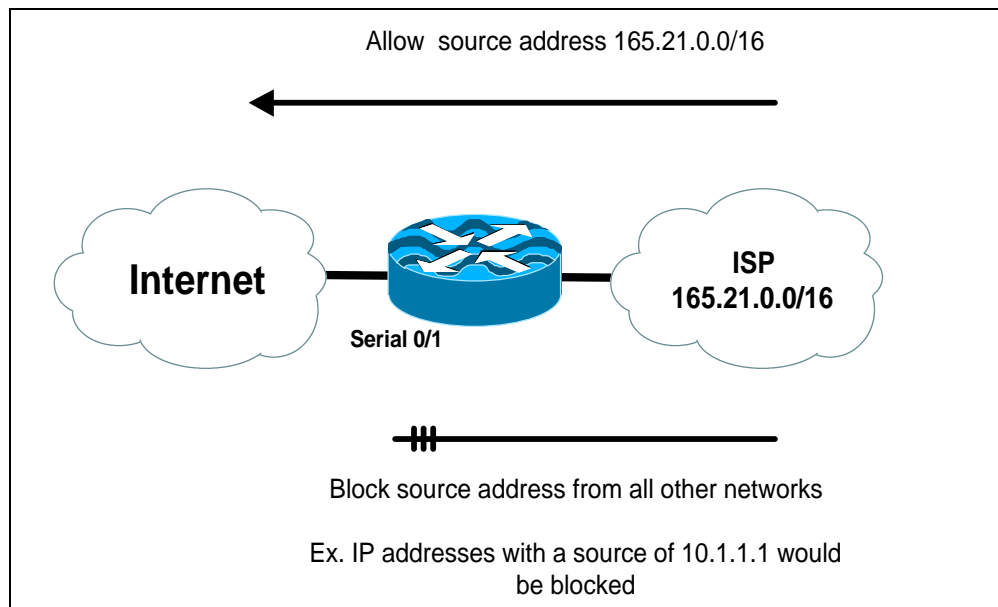


Figure 3 – Egress Filtering

Ingress Filtering – Preventing Reception of Invalid IP Addresses

For ISPs who provide service to end networks, we highly recommend the validation of incoming packets from your clients. This can be accomplished by the use of inbound packet filters on your border routers (Figure 4). For example, if your customer has a network number of 165.21.0.0/16, you should not see any packets coming into your network with 165.21.0.0 as the source address. These packets are attempts at spoofing and should be dropped. The following example shows a sample filter for network 165.21.0.0 with filters for private and rogue routes:

```
access-list 111 deny ip host 0.0.0.0 any log
access-list 111 deny ip 127.0.0.0 0.255.255.255 any log
access-list 111 deny ip 10.0.0.0 0.255.255.255 any log
access-list 111 deny ip 172.16.0.0 0.15.255.255 any log
access-list 111 deny ip 192.168.0.0 0.0.255.255 any log
access-list 111 deny ip 165.21.0.0 0.0.255.255 any log
access-list 111 permit ip any any

interface serial 1/0
 ip access-group 111 in
```

All the “anti spoof”, private address, and rogue filters have *log any* matches. It is not crucial to have this line, but it will help locate the source and extent of the possible probes or attacks.

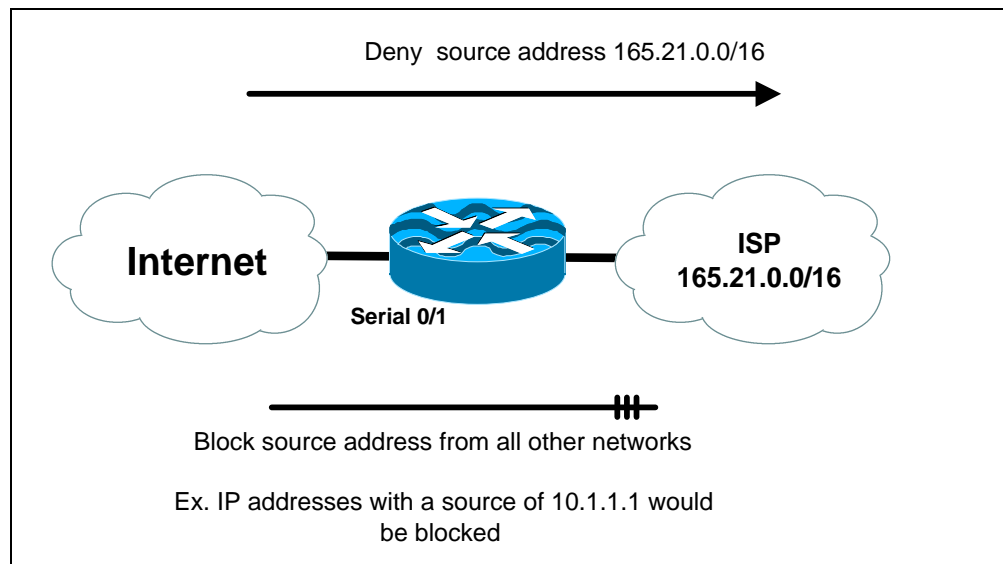


Figure 4 – Ingress Filtering

Unicast RPF – Reverse Path Forwarding¹¹

Unicast Reverse Path Forwarding (RPF) is a feature that can be used in the access network to help prevent Denial of Service (DoS) attacks based on Source IP address spoofing – including the famous SMURF attack. Unicast RPF requires the Cisco Express Forwarding (CEF) mechanism to be enabled. The effect of Unicast RPF is that it stops SMURF attacks (and other attacks that depend on source IP address spoofing) at the ISP's PoP (leased line and dial-up). This protects the ISP's network, its customers' networks, as well as the rest of the Internet (good Netizen operation).

Unicast RPF works by validating the IP source address of the packet against the information provide by CEF's routing information (the FIB – Flow Information Base). Unicast RPF ensures that there is a *reverse path route* to the input interface of the packet. If there is a *reverse path route*, the packet is forward as normal. If there is not a reverse path route, the packet is dropped.

When a packet is received by a router's interface with RPF and ACLs configured, the following occurs:

1. Check input ACLs configured on the inbound interface.
2. Unicast RPF validates the packet has a return path through the inbound interface by checking the CEF table (CEF or dCEF).
3. CEF table (FIB) lookup is carried out for packet forwarding.
4. Output ACLs are checked on outbound interface.
5. Packet is forwarded.

For example, if a customer sent a packet with the source address of 210.210.1.1 from interface FDDI 2/0/0, RPF will check the FIB to see if 210.210.1.1 has a path to FDDI 2/0/0. If there is a matching path, the packet will be forward. If there is no matching path, the packet will be dropped.

Unicast RPF was first supported in 11.1(17)CC CEF images on the RSP7000, 7200 and 7500 platforms. It is not supported in any 11.2 or 11.3 images. Unicast RPF is included in 12.0 on platforms that support CEF. This includes the AS5800 which means that Unicast RFP can be configured on the PSTN/ISDN dial-up interfaces on that platform.

¹¹ Original documentation on Unicast RPF was done by Bruce R. Babcock [bbabcock@cisco.com]

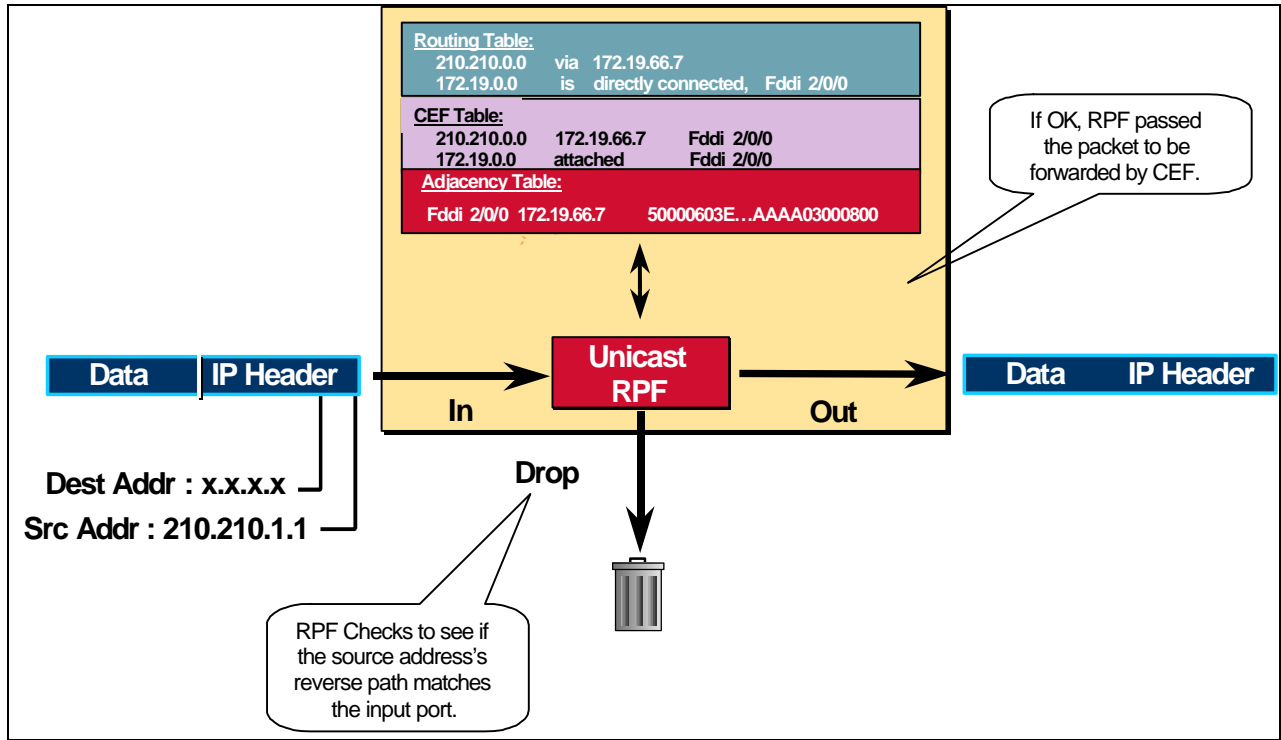


Figure 5 – Unicast RPF validating IP source addresses

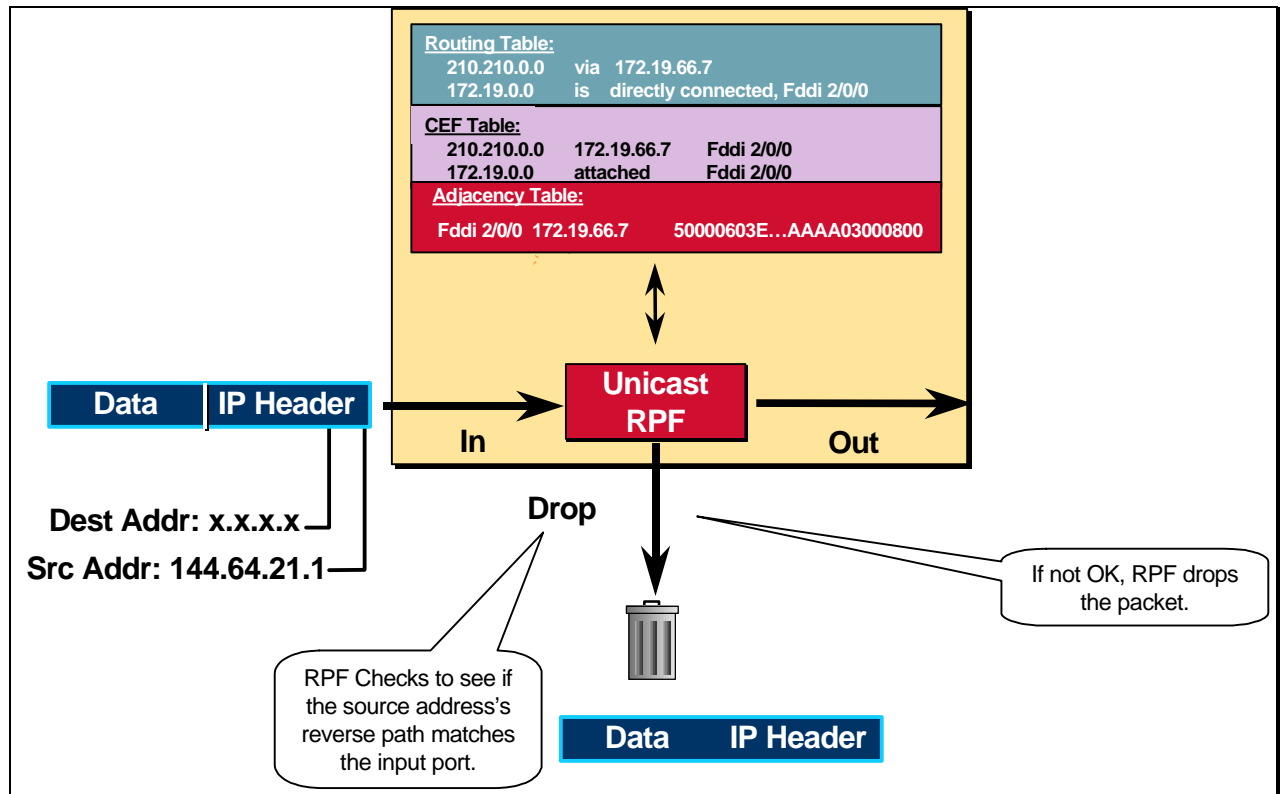


Figure 6 – Unicast RPF dropping packets which fail verification

RPF Configuration Details

To use unicast RPF, enable ‘CEF switching’ or ‘CEF distributed switching’ in the router. There is no need to configure the input interface for CEF switching. This is because unicast RPF has been implemented as a search through the FIB using the source IP address. As long as CEF is running on the router, individual interfaces can be configured with other switching modes. RPF is an input side function that is enabled on an interface or sub-interface supporting any type of encapsulation and operates on IP packets received by the router. **It is very important for CEF to be turned on globally in the router – RPF will not work without CEF.**

Configure RPF on the interface using the following interface command syntax:

```
[no] ip verify unicast reverse-path
```

For example on a leased line aggregation router:

```
ip cef
! or "ip cef distributed" for an RSP+VIP based box
!
interface serial 5/0/0
 ip verify unicast reverse-path
```

For another example, the AS5800 supports CEF in the IOS 12.0. The *interface Group-Async* command makes it even easier to apply Unicast RPF on all the dial-up ports.:

```
ip cef
!
interface Group-Async1
 ip verify unicast reverse-path
```

Use the command *show cef interface [interface]* to verify RPF is operational:

```
ISP-LAB-7505-3#sh cef inter serial 2/0/0
Serial2/0/0 is up (if_number 8)
 Internet address is 169.223.10.2/30
 ICMP redirects are never sent
 Per packet loadbalancing is disabled
 IP unicast RPF check is enabled
 Inbound access list is not set
 Outbound access list is not set
 Interface is marked as point to point interface
 Packets switched to this interface on linecard are dropped to next slow path
 Hardware idb is Serial2/0/0
 Fast switching type 4, interface type 6
 IP Distributed CEF switching enabled
 IP LES Feature Fast switching turbo vector
 IP Feature CEF switching turbo vector
 Input fast flags 0x40, Output fast flags 0x0, ifindex 7(7)
 Slot 2 Slot unit 0 VC -1
 Transmit limit accumulator 0x48001A02 (0x48001A02)
 IP MTU 1500
```

A counter is maintained to count the number of discards due to RPF. The value of the counter will be displayed as part of the output from the command:

```
show ip traffic
```

The RPF drop counter is included in the *IP Statistics* section:

```
ISP-LAB-7505-3#sh ip traffic
IP statistics:
 Rcvd: 1471590 total, 887368 local destination
       0 format errors, 0 checksum errors, 301274 bad hop count
```

Saturday, February 19, 2000

```
0 unknown protocol, 0 not a gateway
0 security failures, 0 bad options, 0 with options
Opts: 0 end, 0 nop, 0 basic security, 0 loose source route
0 timestamp, 0 extended security, 0 record route
0 stream ID, 0 strict source route, 0 alert, 0 other
Frgs: 0 reassembled, 0 timeouts, 0 couldn't reassemble
0 fragmented, 0 couldn't fragment
Bcast: 205233 received, 0 sent
Mcast: 463292 received, 462118 sent
Sent: 990158 generated, 282938 forwarded
Drop: 3 encapsulation failed, 0 unresolved, 0 no adjacency
0 no route, 0 unicast RPF, 0 forced drop
*****
```

Figure 7 – Unicast RPF Drop Counter

RPF packets that are dropped can be captured with a *debug ip cef drops rpf* statement (as of 12.0(4)S but not in 11.1CA or in 11.1CC).

RPF Implementation Notes

RPF has the advantage (over access-lists) when used for IP address spoof prevention in that it dynamically adapts to changes in the dynamic routing tables, including static routes. RPF has minimal CPU overhead and operates a few percent less than CEF/optimum/fast switching rates. RPF has far lower performance impact as an anti-spoofing tool compared with the access-list approach. Access-lists are still required to combat ‘spoofing’ where asymmetric routing is present in the network.

RPF also requires less operational maintenance than traditional approaches which use IP access or extended-access lists. It can be added to the customer’s default configuration on the ISP’s router (remember this will only work if the router has CEF configured).

```
! Configuration template for customer interfaces
description [enter description of interface]
no ip redirect
no ip direct broadcast
no ip proxy-arp
ip verify unicast reverse-path
bandwidth [bandwidth in kbps]
```

RPF is compatible with CEF’s per-packet and per-destination load sharing.

Unicast RPF is best used at the edge of the network for customer network terminations (see Figure 8). Leased line customer aggregation routers are ideal. In this topology, the customer aggregation routers need not have the full Internet Routing Table. It will only need the information on the routing prefixes assigned to the customer¹². Hence information configured or redistributed in the IGP or iBGP (depending on the way you add customer routes into your network) would be enough for Unicast RPF to do its job.

Unicast RPF is not limited to leased line connections. It works equally well on PSTN/ISDN/xDSL customer connections into the Internet. In fact, dial-up connections are reputed to be the greatest source of SMURF based DoS attacks. As long as the Network Access Server (NAS) supports CEF, Unicast RPF will work.

Unicast RPF and Routing Asymmetry

There is some misunderstanding in the ISP community about the perceived dangers of Unicast RPF and routing asymmetry. As long as ISP Engineers mindfully plan which interfaces they activate Unicast RPF on, routing asymmetry is actually not a serious problem. Routers at the edge of an ISP’s network are more likely to have symmetrical reverse paths.

¹² The customer’s assigned IP Block (i.e. routing prefixes) usually is inserted into the ISP’s network in one of several ways. Any of these ways will work as the information is passed to the CEF tables.

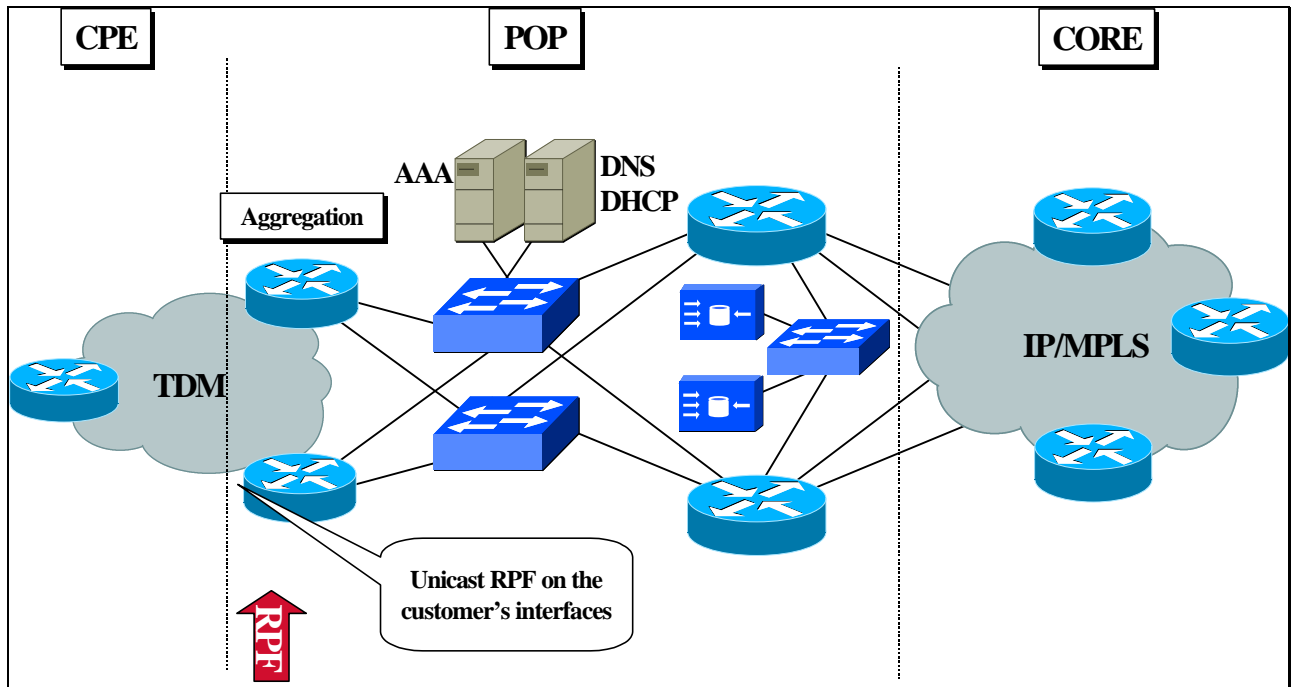


Figure 8 – Unicast RPF applied to Lease Line Customer Connections

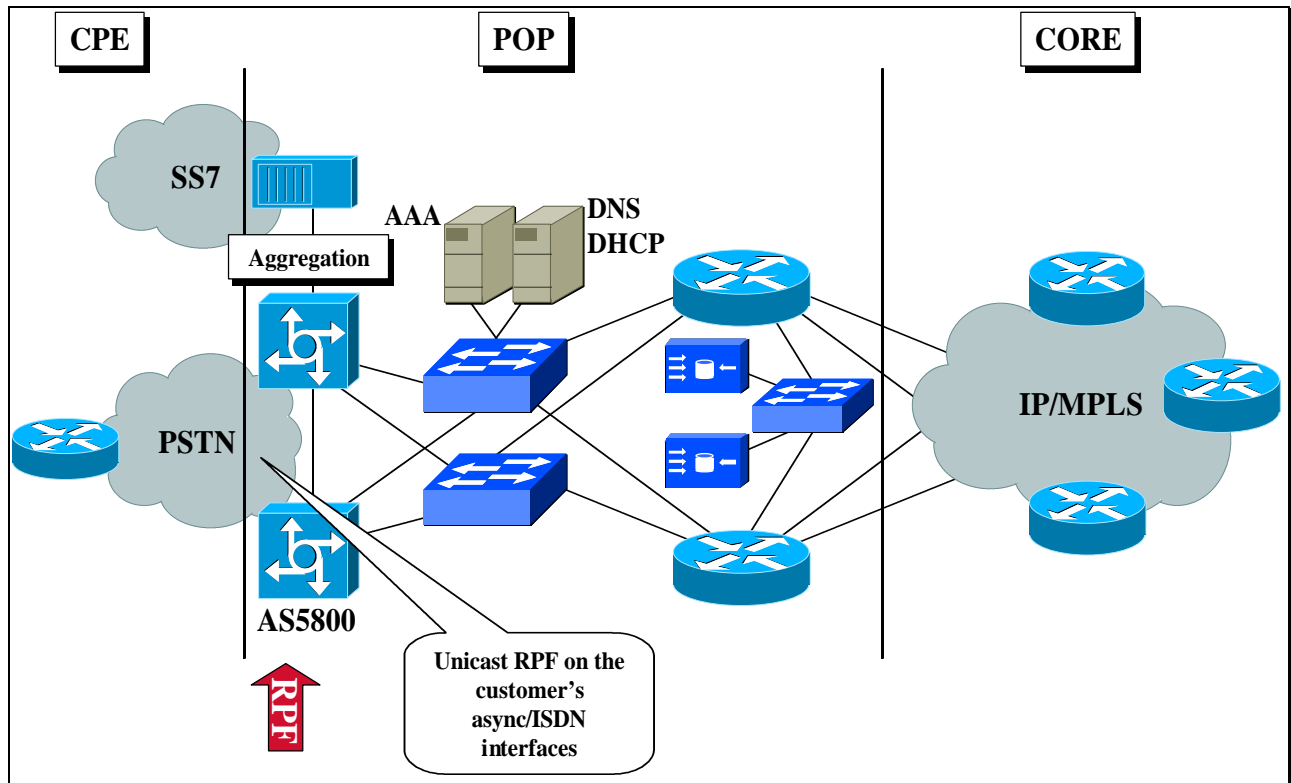


Figure 9 – Unicast RPF applied to PSTN/ISDN Customer Connections

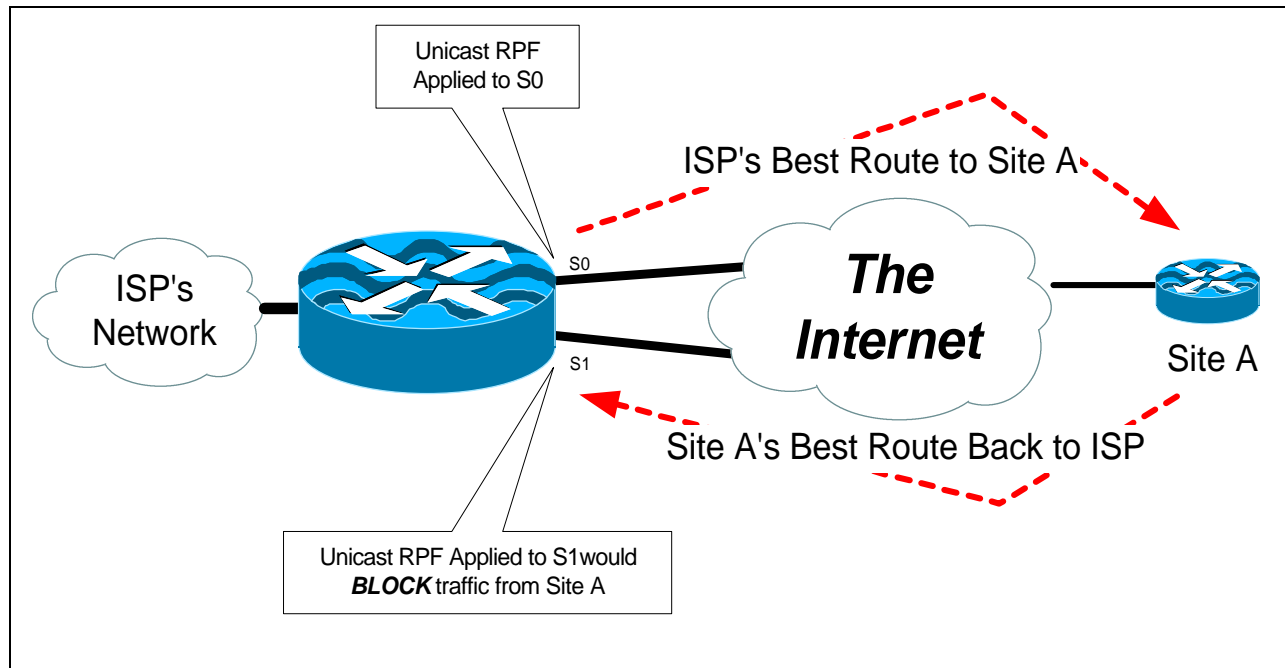


Figure 10 – How asymmetrical routing would not work with Unicast RPF

Routers that are meshed or multi-homed in the core of the ISP's network have no guarantee that the best forwarding path out of the router will be the path selected for packets returning to the router (see Figure 10). Hence, it is not recommended to apply *ip verify unicast reverse-path* where there is a chance of asymmetric routing. It is simplest to place Unicast RPF on the customer edge of an ISP's network (as in Figure 8).

Routing Tables Requirements

Unicast RPF needs proper information in the CEF tables to work properly. This does not mean the router must have the entire Internet routing table. The amount of routing information needed in the CEF tables depends on where Unicast RPF is configured and what functions the router plays in the ISP's network. For example, a router that is a leased line aggregation router for customers only needs the information based on the static routes redistributed into the IGP or iBGP (depending on which technique is used in the network). Unicast RPF would be configured on the customers' interfaces – hence the requirement for minimal routing information. In another scenario, a single-homed ISP can place Unicast RPF on the gateway link to the Internet. The full Internet routing table would be required. This would help protect the ISP from external DoS attacks that use addresses that are not in the Internet routing table.

Unicast RPF Exceptions

Unicast RPF will now allow packets with 0.0.0.0 source and 255.255.255.255 destination to pass so that BOOTP and DHCP function. This feature (CSCdk80591) was added from IOS release 12.0(03.05) (but is not in 11.1CC).

Unicast RPF Example – Putting it all together

The following is an ISP allocated CIDR block 165.21.0.0/16 with both inbound and outbound filters on the upstream interface:

```
ip cef distributed
!
interface Serial 5/0/0
description Connection to Upstream ISP
```

```

ip address XXX.XXX.XXX.XXX 255.255.255.252
no ip redirects
no ip directed-broadcast
no ip proxy-arp
ip verify unicast reverse-path
ip access-group 111 in
ip access-group 110 out
!
access-list 110 permit ip 165.21.0.0 0.0.255.255 any
access-list 110 deny ip any any log
access-list 111 deny ip host 0.0.0.0 any log
access-list 111 deny ip 127.0.0.0 0.255.255.255 any log
access-list 111 deny ip 10.0.0.0 0.255.255.255 any log
access-list 111 deny ip 172.16.0.0 0.15.255.255 any log
access-list 111 deny ip 192.168.0.0 0.0.255.255 any log
access-list 111 deny ip 165.21.0.0 0.0.255.255 any log
access-list 111 permit ip any any

```

Other Considerations

This example used a very simple single homed ISP to demonstrate the concepts of ingress/egress filters. Be mindful that ISPs are usually not single homed (or if they are, they soon become multihomed). Hence, provisions for asymmetrical flows¹³ need to be designed into the filters on the ISP's borders.

Authenticating Routing Protocol Updates

How do you know the routing updates from one of your internal backbone routers *really* came from a neighbouring router on your backbone? You do not – unless neighbour authentication is used. Theoretically, routing information can be spoofed and injected into an ISP's backbone. The horror of seeing a normal 60000 entry routing table jump to 200000 or 500000 entries, then seeing these propagate all over the Internet, has been quietly talked about in the halls of Internet operations meetings. ISPs are strongly encouraged to prevent their routers from receiving fraudulent route updates by configuring neighbour router authentication.

Neighbour router authentication is part of an ISP's total security plan. This section describes what neighbour router authentication is, how it works, and why it should be used to increase overall network security. Documentation details can be found at:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios113ed/113ed_cr/secur_c/scprt5/scrouter.htm

Benefits of Neighbour Authentication

When configured, neighbour authentication occurs whenever routing updates are exchanged between neighbour routers. This authentication ensures that a router receives reliable routing information from a trusted source. Without neighbour authentication, unauthorised or deliberately malicious routing updates could compromise the security of your network traffic. A security compromise could occur if an unfriendly party diverts or analyses your network traffic. For example, an unauthorised router could send a fictitious routing update to convince your router to send traffic to an incorrect destination. This diverted traffic could be analysed to learn confidential information of your organisation, or merely used to disrupt your organisation's ability to effectively communicate using the network. Neighbour Authentication prevents any such fraudulent route updates from being received by your router.

Protocols That Use Neighbour Authentication

Neighbour authentication can be configured for the following routing protocols:

- Border Gateway Protocol (BGP)

¹³ *Asymmetrical Flows* are when the out bound traffic goes out one link and returns via a different link.

Saturday, February 19, 2000

- DRP Server Agent
- Intermediate System-to-Intermediate System (IS-IS)
- IP Enhanced Interior Gateway Routing Protocol (IGRP)
- Open Shortest Path First (OSPF)
- Routing Information Protocol (RIP) version 2

When to Configure Neighbour Authentication

You should configure any router for neighbour authentication if that router meets all of these conditions:

- The router uses any of the routing protocols previously mentioned.
- It is conceivable that the router might receive a false route update.
- If the router were to receive a false route update, your network might be compromised.
- If you configure a router for neighbour authentication, you also need to configure the neighbour router for neighbour authentication.

How Neighbour Authentication Works

When neighbour authentication has been configured on a router, the router authenticates the source of each routing update packet that it receives. This is accomplished by the exchange of an authenticating key (sometimes referred to as a password) that is known to both the sending and the receiving router. There are two types of neighbour authentication used: plain text authentication and Message Digest Algorithm Version 5 (MD5) authentication. Both forms work in the same way, with the exception that MD5 sends a “message digest” instead of the authenticating key itself. The message digest is created using the key and a message, but the key itself is not sent, preventing it from being read while it is being transmitted. Plain text authentication sends the authenticating key itself over the wire.

Note: Plain text authentication is not recommended for use as part of your security strategy. Its primary use is to avoid accidental changes to the routing infrastructure. Using MD5 authentication, however, is a recommended security practice.

CAUTION: As with all keys, passwords, and other security secrets, it is imperative that you closely guard the keys used in neighbour authentication. The security benefits of this feature are reliant upon your keeping all authenticating keys confidential. Also, when performing router management tasks via Simple Network Management Protocol (SNMP), do not ignore the risk associated with sending keys using non-encrypted SNMP.

Plain Text Authentication

Each participating neighbour router must share an authenticating key. This key is specified at each router during configuration. Multiple keys can be specified with some protocols; each key must be identified by a key number. In general, when a routing update is sent, the following authentication sequence occurs:

Step 1: A router sends a routing update with a key and the corresponding key number to the neighbour router. In protocols that can have only one key, the key number is always zero.

Step 2: The receiving (neighbour) router checks the received key against the same key stored in its own memory.

Step 3: If the two keys match, the receiving router accepts the routing update packet. If the two keys did not match, the routing update packet is rejected.

The protocols using plain text authentication (as of IOS 11.3) are:

- DRP Server Agent
- IS-IS

- OSPF
- RIP version 2

MD5 Authentication

MD5 authentication works similarly to plain text authentication, except that the key is never sent over the wire. Instead, the router uses the MD5 algorithm to produce a “message digest” of the key (also called a “hash”). The message digest is then sent instead of the key itself. This ensures that nobody can eavesdrop on the line and learn keys during transmission. These protocols use MD5 authentication:

- OSPF
- RIP version 2
- BGP
- IP Enhanced IGRP

CAR as a SMURF Reaction/Prevention Tool¹⁴

What is a SMURF or FRAG Attack?

The “smurf” attack is a specific Denial of Service (DoS) attack, named after its exploit program. It is a recent category of network-level attacks against hosts. A perpetrator sends a large amount of ICMP echo (ping) traffic to specific IP broadcast addresses. All the ICMP echo packets will have the spoofed source address of a victim. If the routing device delivering traffic to those broadcast addresses performs the IP broadcast to layer 2, then the ICMP broadcast function will be forward to all host on the layer 2 medium (see Figure 11). Most hosts on that IP network will take the ICMP echo request and reply to it with an echo reply. This multiplies the traffic by the number of hosts responding. On a multi-access broadcast network, there could potentially be hundreds of machines to reply to each packet.

The “smurf” attack’s cousin is “fraggle”, which uses UDP echo packets in the same fashion as the ICMP echo packets; it was a simple re-write of the “smurf” programme. Currently, the systems most commonly hit are Internet Relay Chat (IRC) servers and their providers.

There are two parties who are hurt by this attack:

- The intermediary (broadcast) devices – called the “amplifiers”
- The spoofed address target – the “victim”

The victim is the target of the large amount of traffic that the amplifiers generate.

Consider a scenario which paints a picture of the dangerous nature of this attack. Assume a co-location switched network with 100 hosts, and that the attacker has a T1 circuit. The attacker sends, say, a 768kb/s stream of ICMP echo (ping) packets, with a spoofed source address of the victim, to the broadcast address of the “bounce” or amplifier site. These ping packets hit the bounce site’s broadcast network of 100 hosts; each of them takes the packet and responds to it, creating 100 ping replies out-bound. If you multiply the bandwidth, you will see that 76.8 Mbps is generated outbound from the “bounce site”. Because of the spoofed source address of the originating packets, these reply packets are then directed towards the victim.

Passive SMURF Defences

All the various tools listed previously are effective in minimising SMURF on the Internet. The top three are listed below. Please review each section for details.

¹⁴ This section is a edited version of Craig A. Huegen’s work on SMURF and FRAG protection. For the latest information, please refer to Craig’s page at <http://www.quadrunner.com/~chuegen/smurf.txt> Criag can be contacted at chuegen@cisco.com or chuegen@quadrunner.com.

Saturday, February 19, 2000

- Interface Services (*no ip directed broadcast*)
- Egress and Ingress Filtering
- Unicast RPF – Reverse Path Forwarding

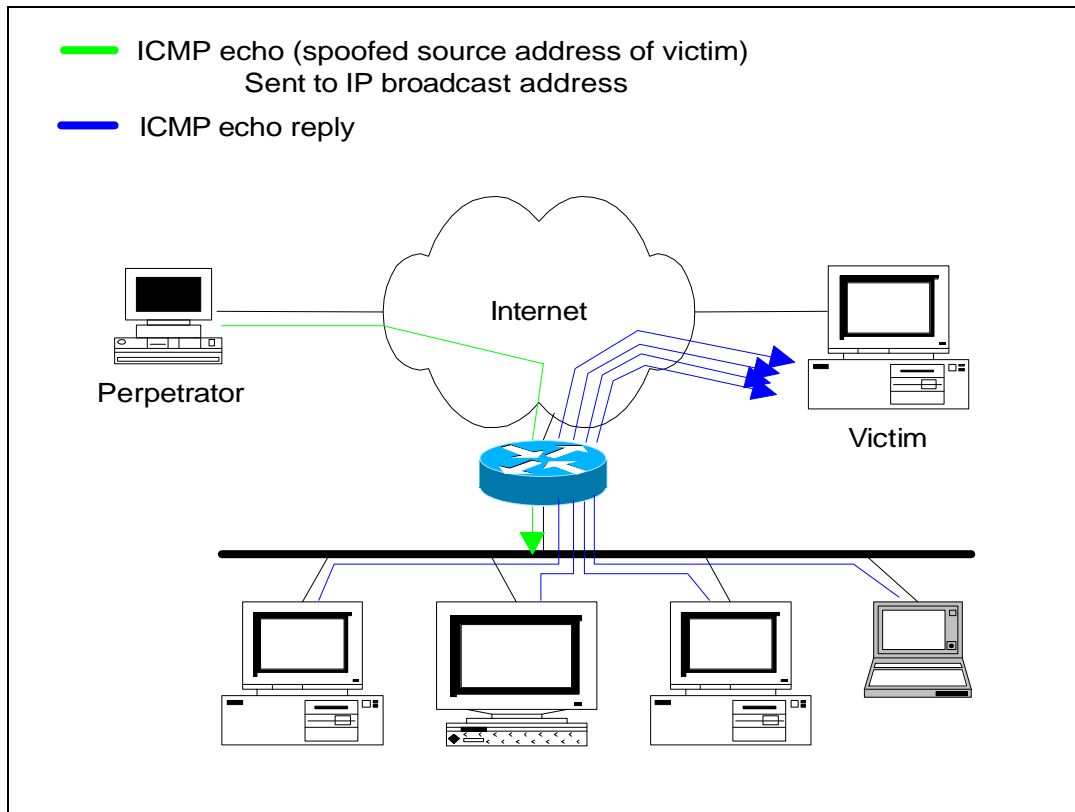


Figure 11 – How SMURF uses amplifiers

Active SMURF Defences

Active defences are tools/techniques/procedures executed when during attacks. They are used to limit and/or block the attack in progress. In many instances these tools will have an effect on other Internet applications and services – yet, the trade off is between no Internet services and limited disruption. It is highly recommended that the ISP document and train staff on the use of these tools. In that way, the ISP's NOC can quickly respond to an attack in progress.

Cisco's IOS has security tools to cover the broad range of networking. New security tools are being added all the time. Yet many of these tools are situation specific. Some tools were written to help enterprise networks – other tools/techniques are more suited for ISP environment. The following sections list the known tools.

Rate Limiting with CAR

It is an inevitable fact of life on the Internet that every ISP is bound to experience a Denial of Service (DoS) attack. Hence, ISPs should have tools and procedures in place to respond to these DoS attacks. Committed Access Rate (CAR) is one such tool that an ISP can use react to DoS attacks.

CAR is a functionality that works with Cisco Express Forwarding, found in 11.1CC and releases from 12.0. It allows network operators to rate limit certain types of traffic to specific sources and/or destinations. The main advantage of CAR is that it can work on packets as they arrive on the router's interface – dropping/rate limiting the DoS flow before any other packet processing.

The following URLs provide details on CAR:

Committed Access Rate (CAR)

<http://www.cisco.com/warp/public/732/Tech/car/index.html>

Configuring Committed Access Rate

http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/qos_c/qcpart1/qccar.htm

Example 1 – A provider has filtered its IRC server from receiving ICMP echo-reply packets in order to protect it. Now many attackers are going after the customer’s devices in order to fill some network segments.

The provider above chose to use CAR in order to limit all ICMP echo and echo-reply traffic received at the borders to 256 Kbps. An example follows:

```
! traffic we want to limit
access-list 102 permit icmp any any echo
access-list 102 permit icmp any any echo-reply
! interface configurations for borders
interface Serial3/0/0
  rate-limit input access-group 102 256000 8000 8000 conform-action transmit exceed-action drop
```

This limits ICMP echo and echo-reply traffic to 256 Kbps with a small amount of burst. Multiple “rate-limit” commands can be added to an interface in order to control other kinds of traffic as well.

The command “*show interface [interface-name] rate-limit*” will show the statistics for rate-limiting; “*clear counters [interface-name]*” will clear the statistics for a fresh look.

Example 2 – Use CAR to limit TCP SYN floods to particular hosts -- without impeding existing connections. Some attackers have started using very high streams of TCP SYN packets in order to harm systems.

This example limits TCP SYN packets directed at host 10.0.0.1 to 8 kbps or so:

```
! We don't want to limit established TCP sessions -- non-SYN packets
access-list 103 deny tcp any host 10.0.0.1 established
! We do want to limit the rest of TCP (this really only includes SYNs)
access-list 103 permit tcp any host 10.0.0.1
! interface configurations for network borders
interface Serial3/0/0
  rate-limit input access-group 103 8000 8000 8000 conform-action transmit exceed-action drop
```

ROUTING

The Routing chapter of this whitepaper assumes that the ISP Engineer has a working understanding of the core routing protocols used in the Internet. If not, please refer to the following on-line whitepapers and/or publications:

Internet Routing Architectures, New Riders Publishing (Cisco Press). ISBN 1-56205-652-2. Author: Bassam Halabi.

Using the Border Gateway Protocol for Interdomain Routing. Available on the *Cisco Documentation CD* or publicly on-line via Cisco Connection On-Line (CCO):

<http://www.cisco.com/univercd/data/doc/cintrnet/ics/icsbgp4.htm>

Internetworking Technology Overview. On-line whitepapers and tutorials on the essentials of routing and switching. Available on the *Cisco Documentation CD* or publicly on-line via Cisco Connection On-Line (CCO):

<http://www.cisco.com/univercd/data/doc/cintrnet/75818.htm>

Technology Information and Whitepapers. Key references and practical internetworking examples. Available on the *Cisco Documentation CD* or publicly on-line via Cisco Connection On-Line (CCO):

<http://www.cisco.com/univercd/data/doc/cisintwk.htm>

Hot Standby Routing Protocol

A new feature in recent versions of IOS is HSRP (see RFC2281 – <http://info.internet.isi.edu/in-notes/rfc/files/rfc2281.txt>). This feature is especially useful on LANs within the ISP's backbone, for example for network servers, non-Cisco access servers, and hosted servers.

The motivation for this protocol is to support the need for a default gateway on LAN networks when there are two gateway routers providing connectivity to the wider network and Internet. Only routers tend to support the full set of routing protocols. Computer workstations in majority run variants of Unix or Windows95/NT, for which there are either no or minimally functional routing software. Configuring something like the public domain software *gated* or the vendor's own software to perform a dynamic routing function is usually a poor and unreliable compromise. The best solution is to configure a static default route and use HSRP.

Figure 12 shows a typical LAN with two routers used to connect to the ISP backbone. To implement HSRP, the configuration required for these two routers looks something like this:

```
Router 1:
  interface ethernet 0/0
  description Server LAN
  ip address 169.223.10.1 255.255.255.0
  standby 10 ip 169.223.10.254
```

```
Router 2:
  interface ethernet 0/0
  description Service LAN
  ip address 169.223.10.2 255.255.255.0
  standby 10 priority 150
  standby 10 preempt
  standby 10 ip 169.223.10.254
```

The two routers have their LAN IP addresses conventionally defined in the configuration above. However, another IP address has been defined, in the command line *standby 10 ip 169.223.10.254*. This address is the address of the virtual default gateway defined on the LAN. All the systems on the LAN, apart from **router1** and **router2** use this address as the "default route". Router2 has a standby priority of 150, higher than the default of 100. Therefore **router2** will be the default gateway at all times unless it is unavailable (i.e. down). The *preempt* directive tells **router1** and **router2** that **router2**

should be used as default gateway whenever possible. For example, if **router2** were temporarily out of service, it would take over from **router1** when it is returned to normal operation.

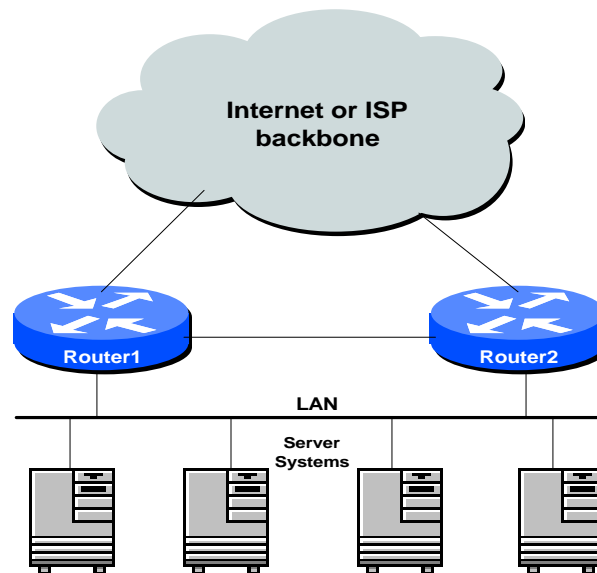


Figure 12 – Dual gateway LAN

A problem here is that all the outbound traffic goes through **router2**, whereas inbound traffic may be shared between the two. Some ISPs like to share outbound traffic between the two routers, and this is achieved by setting up two standby groups as in the following example:

```
Router 1:
interface ethernet 0/0
description Server LAN
ip address 169.223.10.1 255.255.255.0
standby 10 ip 169.223.10.254
standby 11 priority 150
standby 11 preempt
standby 11 ip 169.223.10.253
```

```
Router 2:
interface ethernet 0/0
description Service LAN
ip address 169.223.10.2 255.255.255.0
standby 10 priority 150
standby 10 preempt
standby 10 ip 169.223.10.254
standby 11 ip 169.223.10.253
```

Provided the ISP shares the default route configuration on the servers between the two virtual default gateways he should see a reasonably balanced sharing of traffic on the two outbound links.

CIDR Features

All network devices connected to the Internet should be Classless Inter-Domain Routing (CIDR) compliant. Cisco routers are made CIDR compliant if the two commands are entered:

```
ip subnet-zero
ip classless
```

Saturday, February 19, 2000

All Cisco routers connected to the Internet must have these commands turned on. IOS releases from 12.0 have these commands turned on by default. It makes good sense in the ISP operations world never to rely on any equipment vendor's defaults. Defaults are for enterprise and other networks, not the specialised and public ISP Backbones. (See BGP section for BGP requirements.)

Selective Packet Discard

When a link goes to a saturated state, the router will drop packets. The problem is that the router will drop any type of packets – including routing protocol packets. Selective Packet Discard (SPD) will attempt to toss non-routing packets instead of routing packets when the link is overloaded. In releases 11.1CA and 11.1CC, the configuration command:

```
ip spd enable
```

will switch on SPD. Selective Packet Discard is enabled by default on 11.2(5)P and more recent releases – if desired, though not recommended, its state can be toggled with the commands:

```
[no] spd enable
```

The basic idea behind Selective Packet Discard (SPD) is this: if we mark all BGP and IGP packets as being “important” and prefer these packets over others, we should process a larger percentage of the packets that will allow routing and, consequently, keep the BGP and IGP sessions stable.

SPD images have the following new knobs – some of these are hidden from the IOS configuration helper, but have been made known to the ISP community. (Note the difference between 11.1CC and 12.0 release versions of the commands.)

```
[no] ip spd enable      ! for 11.1CC  
[no] spd enable        ! for 12.0+
```

turns SPD on/off, default is on

```
ip spd headroom        ! for 11.1CC  
spd headroom           ! for 12.0+
```

Default value is 100. Specifies how many high-precedence packets we will enqueue over the normal input hold queue limit. This is to reserve room for incoming high precedence packets.

```
[no] ip spd queue {min-threshold | max-threshold} <n>
```

Sets lower and upper ip process-level queue thresholds for SPD. With SSE based SPD, lower precedence packets are randomly dropped when the queue size hits min-threshold. The drop probability increases linearly with the queue size until max-threshold is reached, at which point all lower precedence packets are dropped. For regular SPD, lower precedence packets are dropped when the queue size reaches min-threshold. Defaults are 50 and 75, respectively. These values were not based on real life experience and may need some tuning.

New “show ip spd” exec command:

```
Current mode: normal.  
Queue min/max thresholds: 73/74  
IP normal queue: 0, priority queue: 0.  
External SPD enabled.
```

(Shows SPD mode, current and max size of IP process level input queue, and status of external (SSE) SPD. SPD mode will be one of disabled, normal, random drop, or full drop. The priority queue is where high-precedence packets go.)

“*show interface switching*” has some extra information too:

```
gmajor#sh int eth 3/0 switching
Ethernet3/0
  Throttle count:          0
SPD Flushes      Fast          0      SSE      542019
  (separates RP/SP flushes)
SPD Priority     Inputs:      123     Drops:      0
  (priority packets received and dropped due to exceeding
  headroom threshold.)
```

Note that “*switching*” is a hidden command, and needs to be entered in full to be recognised by the command parser.

IP Source Routing

The Cisco IOS software examines IP header options on every packet. It supports the IP header options Strict Source Route, Loose Source Route, Record Route, and Time Stamp, which are defined in RFC 791. If the software finds a packet with one of these options enabled, it performs the appropriate action. If it finds a packet with an invalid option, it sends an ICMP Parameter Problem message to the source of the packet and discards the packet.

IP has a feature that allows the source IP host to specify a route through the IP network. This feature is known as source routing. Source routing is specified as an option in the IP header. If source routing is specified, the software forwards the packet according to the specified source route. The default is to perform source routing.

Some ISPs do not want their customers to have access to source routing, hence it is turned off at the customer edge. Some ISPs like to have source routing available to troubleshoot their and other networks – especially when routing has broken inside one of those networks. Other ISPs require source routing to be turned on when peering with ISPs.

As a general rule of thumb, *if you are not using ip source routing, turn it off*. IP source routing is a well-known security vulnerability used in attacks against a system.

```
no ip source-route
```

Configuring Routing Protocols

This section examines the most efficient and effective ways of configuring Internal Gateway Protocols (IGPs) and BGP to give greatest scalability in IOS.

There are essentially three IGPs which are used by ISPs. These are ISIS (Intermediate System to Intermediate System), OSPF (Open Shortest Path First) and EIGRP (Enhanced Internal Gateway Routing Protocol). The former are industry standards (ISIS was developed by OSI now being enhanced by the IETF, OSPF was developed by the IETF), while EIGRP is an enhancement of IGRP developed by Cisco since the late ‘80s.

BGP version 4 has been used since 1993 as the routing protocol for autonomous systems to exchange prefixes with each other. BGP4 is the classless version of the Border Gateway Protocol and has seen many enhancements over the last few years as the Internet grows larger and larger.

Router ID

The router identifier in an IGP is chosen from the loopback interface of the router, if configured. If more than one loopback interface is configured with an IP address, the router identifier is the highest of those IP addresses. If the loopback is not configured, the highest IP address configured on the router at the time the IGP was started is used. ISPs prefer stability, so the loopback interface is usually configured and active on most ISP routers.

Saturday, February 19, 2000

The router ID is also used as the last step of the BGP path selection process. Another reason to ensure that the loopback interface is configured and has an IP address – if there is no loopback, the router ID is the highest IP address configured on the box at the time the BGP process was started.

Choosing an IGP

A general discussion on the choice of IGPs is beyond the scope of this document. However, the choice of IGP generally seems to be made on the basis of experience, as technically there is little to choose between the three for most practical purposes. Those engineers with a strong ISIS background will always choose ISIS. Those with a strong OSPF background will always choose OSPF. The rule of thumb seems to be that beginners to interior routing choose EIGRP as it is easy to get started. However, OSPF is a better choice as it forces good IGP design to ensure that the network will scale. And those who are very experienced tend to choose ISIS on Cisco routers as it has “more scalability knobs” than the other two IGPs.

Configuring an IGP

How to configure an IGP is also beyond the scope of this document too. There are many good documents available on CCO which can help with IGP design. However, listed below are some good IOS tips available which will help ISPs whichever IGP is chosen. There is no harm in repeating the ISP engineer’s well known design tips though:

- There are three types of prefixes:
 1. Access network prefixes
 2. Infrastructure prefixes
 3. External prefixes
- IGPs carry **infrastructure** prefixes only
- Access network prefixes are not part of the infrastructure – use **ip unnumbered** if possible
- Customer prefixes are **never** carried in an IGP – BGP is designed for this
- IGPs are kept small for best convergence speed – use a good address plan and summarisation
- iBGP and eBGP prefixes are **never ever** distributed into an IGP
- IGP prefixes are **never ever** distributed into BGP

Putting Prefixes into the IGP

The Network Statement

There are at least three possible ways of inserting prefixes into the IGP. However, ISPs aim to find the most efficient method, and one which will give greatest scalability of their network. The most efficient, scalable, and safe method is to use the **network** statement to cover the infrastructure addresses which will have an active IGP running over them and/or require to be carried in the IGP. A sample configuration might be:

```
interface loopback 0
 ip address 220.220.16.1 255.255.255.255
!
interface serial 0/0
 ip address 220.220.17.1 255.255.255.252
!
interface serial 1/0
 ip unnumbered loopback 0
!
router ospf 100
 network 220.220.16.1 0.0.0.0 area 0
 network 220.220.17.0 0.0.0.3 area 0
 passive-interface loopback 0
 passive-interface serial 1/0
!
```

Notice the use of the passive-interface command to disable OSPF neighbour discovery on any interfaces which do not have connected neighbours or don’t need to run a routing protocol.

It is not good practice, and indeed strongly discouraged to redistribute anything into an interior routing protocol. It is rarely required.

Redistribute Connected into an IGP

Using the **redistributed connected** command takes the prefixes assigned to every connected interface and injects them into the IGP. Apart from creating an external network type which, for example, in OSPF is carried across all areas, the router periodically has to examine the Routing Information Base (RIB) to see if there are any changes to the connected state. This takes extra CPU cycles. Additionally, any link state changes are passed into the IGP, something which may not be desirable especially in the case of link addresses going to external connections. In the OSPF example, the external link state change is heard over the whole network.

Another problem is that redistributing connected interfaces into an IGP covers **all** connected interfaces on a routing device. Some addresses maybe aren't desirable in an IGP, so ISPs would then have to add a route-map to apply an access-list filter to the redistribution process. Which could add something else to go wrong – access-lists can be deleted by accident, resulting in prefixes being leaked into the IGP. Again, undesirable.

Conclusion: don't use redistribute connected.

Redistribute Static into an IGP

Using the **redistributed static** command takes all static routes and injects them into the IGP. Apart from creating an external network type which, for example, in OSPF is carried across all areas, the router periodically has to examine the Routing Information Base (RIB) to see if there are any additions to or deletions from the static route configuration. This takes extra CPU cycles. And if the next hop to which the static route points disappears, the static route is withdrawn, with the resulting withdrawal from the IGP. (Ofcourse, the recently introduced permanent static route is designed to workaround this problem, but the problem is best avoided in the first place!)

The other problem with redistributing static routes into an IGP is that it covers **all** static routes configured on a routing device. Some prefixes maybe aren't desirable in an IGP, resulting in a situation similar to that noted in the “redistributed connected” case.

Conclusion: don't use redistribute static.

Redistribute <anything> into an IGP

Conclusion: don't. ☺

IGP Summarisation

It is good practice, and aids scalability, to ensure that summarisation is implemented wherever possible. IGP's work most efficiently with as few prefixes as possible in the routing protocol. Small IGP's converge more quickly, ensuring rapid network healing in case of link failures. How to configure summarisation is different for each protocol – consult the documentation for those. For example, OSPF uses the “**area <n> range**” OSPF subcommand.

IGP Adjacency Change Logging

Enable neighbour state logging in each IGP. This means that it becomes easier to find out about neighbour states, reasons for state changes, etc. In OSPF's case, logging is enabled by the OSPF subcommand “**ospf log-adjacency-changes**”.

Log messages are sent to the router console, and wherever else the logging output has been configured. For most ISPs, this would be a Unix syslog server – see the section discussing router logs for how to configure this. The typical output would

Saturday, February 19, 2000

be similar to what is given below – this example shows the activation of a neighbour adjacency with a router connected to serial3/2.

```
Nov 10 03:56:23.084 AEST: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.2.2 on Serial3/2 from DOWN to
INIT, Received Hello
Nov 10 03:56:32.620 AEST: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.2.2 on Serial3/2 from INIT to
2WAY, 2-Way Received
Nov 10 03:56:32.620 AEST: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.2.2 on Serial3/2 from 2WAY to
EXSTART, AdjOK?
Nov 10 03:56:32.640 AEST: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.2.2 on Serial3/2 from EXSTART
to EXCHANGE, Negotiation Done
Nov 10 03:56:32.676 AEST: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.2.2 on Serial3/2 from EXCHANGE
to LOADING, Exchange Done
Nov 10 03:56:32.676 AEST: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.2.2 on Serial3/2 from LOADING
to FULL, Loading Done
```

Putting Prefixes into BGP

The Network Statement

As with IGPs, there are at least three possible ways of inserting prefixes into BGP. The safest method is to use one **network** statement per prefix which is to be injected into BGP. A sample configuration might be:

```
interface loopback 0
 ip address 220.220.16.1 255.255.255.255
!
interface serial 1/0
 ip unnumbered loopback 0
!
router bgp 100
 no synchronization
 network 220.220.18.0 mask 255.255.252.0
!
 ip route 220.220.18.0 255.255.252.0 serial 1/0
```

It is not good practice, and indeed strongly discouraged to redistribute any interior routing protocol into BGP. It is never required and in the past has led to many serious accidents on the global Internet.

Redistribute Connected into BGP

Using the **redistributed connected** command takes the prefixes assigned to any connected interface and injects them into BGP. One disadvantage, as with IGP, is that the router periodically has to examine the Routing Information Base (RIB) to see if there are any changes to the connected state. This takes extra CPU cycles. Additionally, any link state changes are passed into BGP generating a route flap on the Internet routing table, wasting considerable CPU cycles on all routers in the Internet which have a view of this prefix.

Another problem is that redistributing connected interfaces into BGP covers **all** connected interfaces on a routing device. If some addresses aren't desirable in BGP, ISPs would need to add a route-map to apply an access-list filter to the redistribution process. Which could add something else to go wrong – access-lists can be deleted by accident, resulting in prefixes being leaked into BGP. Again, very undesirable.

Conclusion: don't use redistribute connected.

Redistribute Static into BGP

Using the **redistributed static** command takes all static routes and injects them into BGP. Again the router periodically has to examine the Routing Information Base (RIB) to see if there are any additions to or deletions from the static route configuration. This takes extra CPU cycles. And if the next hop to where the static route points disappears, the static route is withdrawn, with the resulting withdrawal from BGP. A withdrawal from BGP results in a route flap, as above. (As in the

IGP case, the permanent static route is designed to workaround this issue, but the problem is best avoided in the first place!)

The other problem with redistributing static routes into BGP is that it covers **all** static routes configured on a routing device. Some prefixes maybe aren't desirable in BGP (eg the RFC1918 prefixes), so ISPs would add a route-map to apply an access-list filter to the redistribution process. This can potentially lead to the situation discussed in the case of “**redistribute connected**” as per above.

Some situations demand **redistribute static**, especially where communities are attached to the prefixes inserted into the BGP process. This appears to be the preferred process amongst ISPs as they find it easier to manage then applying a prefix matching route-map to individual network statements. The following example is very typical, on a customer aggregation or edge device:

```
router bgp 65534
 redistribute static route-map static-to-bgp
 !
 route-map static-to-bgp permit 5
  match ip address prefix-list netblock-to-bgp
  set community 65534:1234
  set origin igp
 !
 ip prefix-list netblock-to-bgp permit 220.220.0.0/16 le 30
 !
 ip route 220.220.1.16 255.255.255.224 serial 0/0
```

which basically sets community of 65534:1234 on all the ISP's customer prefixes. There are dangers with this – the prefix list could be “blown away”, but the ISP has to weigh up this danger with the administrative inconvenience/overhead of setting communities, etc, on a per network statement basis.

Conclusion: don't use redistribute static unless you fully understand why you are doing it.

Redistribute <anything> into BGP

Conclusion: don't. ☺

BGP Features and Commands

BGP is the heart of the Internet. It is the essential protocol that keeps it all *glued* together. Yet, because the only thing you can guarantee on the Internet is change, BGP needs consistent updates with new features and functionality to help ISPs manage and scale their networks. This section provide early documentation and configuration notes for ISP Network Engineers. It is not a replacement for the CCO documentation (<http://www.cisco.com/univercd/home/home.htm>), yet most of these will be documented here long before they are available on CCO.

There are several features and commands that ISPs commonly use and/or are new to the Cisco's implementation of BGP.

```
update-source loopback 0
 ip bgp-community new-format
 no synchronisation
 bgp dampening
 no auto-summary
 bgp neighbor authentication
 bgp neighbor maximum-prefix
 bgp neighbor soft-reconfiguration
 bgp neighbor shutdown
 bgp neighbor next-hop-self
 bgp log-neighbor-changes
 no bgp fast-external-fallover
 bgp peer-groups
```

Saturday, February 19, 2000

```
bgp neighbor x.x.x.x prefix-list <name> in|out
```

We will cover these features/commands briefly in this document. ISP engineers need to understand when to use these commands on their backbones. Hence, detailed examples on how these commands are used can be found in *Using the Border Gateway Protocol for Interdomain Routing* (<http://www.cisco.com/univercd/data/doc/cintrnet/ics/icsbgp4.htm>)

Stable iBGP Configuration

An ISP's backbone should be built using an IGP to carry internal infrastructure addressing, and iBGP to carry customer networks and Regional Registry assigned aggregates. There is the obvious distinction in services and function but more importantly, IGP's converge faster than iBGP, and respond to changes in conditions (physical link status) more quickly. This design is deployed by many ISPs.

A common error is to use IP addresses of ethernet or FDDI interfaces as the remote peer addresses. There may be no issue when the remote address is active, but as soon as the interface goes down, the peering is torn down as the address is no longer reachable. Also, most ISP backbone designs have routers connected by at least two interfaces to the rest of their network. It would be a unfortunate to have the iBGP peering go down if one WAN link had gone down when in fact the router is perfectly accessible via another connection.

The design using two exit paths per router should be taken advantage of. This is done by using the loopback interface, a "real" interface but without any physical connectivity. The interface always exists, and can only be shut down by the IOS configuration command. Choosing the loopback interface and assigning it a host IP address (/32) guarantees a more stable iBGP, no matter the underlying physical network issues which are taken care of by the IGP.

```
hostname gateway1
!
interface loopback 0
  ip address 215.17.1.34 255.255.255.255
!
router bgp 200
  neighbor 215.17.1.35 remote-as 200
  neighbor 215.17.1.35 description iBGP with CR2
  neighbor 215.17.1.35 update-source loopback 0
  neighbor 215.17.1.36 remote-as 200
  neighbor 215.17.1.36 description iBGP with CR3
  neighbor 215.17.1.36 update-source loopback 0
!
```

Routers 215.17.1.35 and 215.17.1.36 see the BGP updates coming from the address 215.17.1.34 of gateway1. Likewise, gateway1 hears the BGP updates coming from 215.17.1.35 and 215.17.1.36 (loopback interfaces configured on those routers). This configuration is independent of the backbone network design, and is not dependent on the physical connectivity of the routers.

The loopback interfaces on an ISP network are usually addressed out of one block of address space, with each loopback address carried around in the ISP's IGP as a /32 address. The reasons for assigning loopback interface addresses out of one block will become apparent later in this appendix.

Notice the use of the "description" command. This is new in 11.1CC and 12.0 software, allowing a description of the BGP peering to be entered into the router configuration. More online documentation!

Note that for eBGP configuration, IP addresses of **real** interfaces are generally used. The reason for this is that there is no IGP running between ISP backbones or ASes. eBGP peering routers have no way of finding out about external networks apart from the other end of a point-to-point WAN link which will be linking the two together.

BGP Auto Summary

In IOS auto-summarisation will be turned on by default. This feature will automatically summarise subprefixes to the classful network boundaries when crossing classful network boundaries. The IPv4 Internet Registries¹⁵ are now allocating from the former Class A space – an ISP today would more likely be allocated /18 IPv4 address from what used to be the class A space. BGP’s default behaviour would be to take that /18 and advertise a /8. Without the BGP command *no auto-summary*, BGP will auto-summarise the /18 into a /8. This will cause at least confusion on the Internet, but worse potentially “attracting” other service providers’ unroutable traffic to the local backbone, with due consequences on circuit and systems loading.

Example: An ISP was allocated 24.10.0.0/18. The ISP would sub-allocate this /18 for their customers. The ISP would want to advertise the /18 to the Internet. BGP’s default behaviour would be to *auto summarize* the /18 into the **classful** boundary – 24.0.0.0/8 – the old class A. The problem is that other ISPs are also getting /18 allocations from the IPv4 Registry.

In today’s classless Internet world where the former class A space is being efficiently carved up and sub-allocated, ISP and Enterprise backbones which use BGP need to use **no auto-summary**.

BGP Synchronisation

BGP does not advertise a route before all routers within the AS have learned about the route via an IGP. In some cases, you might want to disable synchronisation. Disabling synchronisation allows BGP to converge more quickly, but it might result in dropped transit packets.

You can disable synchronisation if one of the following conditions is true:

- Your AS does not pass traffic from one AS to another AS.
- All the transit routers in your AS run BGP.

Since many ISPs run iBGP across their backbone, BGP synchronisation is turned off to gain the benefits for faster convergence.

BGP Community Format

Many ISPs make extensive use of BGP communities for routing policy decision making. A bgp community tutorial is beyond the scope of this document, but ISP engineers should be aware of two formats supported in IOS. The original format for a community number took the form of a 32-bit integer. More recently, this format was redefined as two 16-bit integers separated by a colon as per the BGP standard. The first 16-bit number is accepted as being the ISP’s AS (because community numbers are exportable between ASes with the “*send-community*” bgp directive). The second 16-bit number is used to represent different policies the ISP wishes to implement. Note that there are some standard communities defined as common or current practice – these are documented in RFC1998.

The configuration command is:

```
ip bgp-community new-format
```

and it converts communities from looking something like 13107210 to a more human friendly 200:10! It should be noted that BGP does not care which format is used internally – the field is still 32 bits in total. This new format is purely for appearance and practicality purposes only.

BGP Neighbour Shutdown

¹⁵ APNIC, RIPE, and ARIN

Saturday, February 19, 2000

A new feature introduced into 11.1CC and 12.0 software is the ability to shutdown a BGP peering without actually removing the configuration. Previously the only way to disable a BGP peering was to delete the configuration from the router. This is very disruptive to the router's functioning, and it increases the likelihood of making mistakes when reinstating the configuration at a later stage.

A neighbouring peering is shutdown with the command example:

```
router bgp 200
  neighbor 169.223.10.1 shutdown
```

To reinstate the peering once the problem or reason for shutdown has been removed, simply enter the opposite command:

```
router bgp 200
  no neighbor 169.223.10.1 shutdown
```

All users of BGP are encouraged to use this command rather than deleting the configuration – it greatly enhances the ease and reliable operation of the network.

BGP Soft-Reconfiguration

A new feature in 11.1CC, 11.2P and 12.0 software is the BGP soft reconfiguration capability. Normally when an ISP requires to change the policy in a BGP peering, the peering itself has to be torn down so that the new policy can be implemented. For peerings exchanging a number of routes in the Internet, this can be extremely disruptive, putting load on the CPU of both routers involved, and resulting in a routing “flap” through the backbone as the ISP's network announcements are withdrawn, and then reinstated.

The alternative is soft-reconfiguration. With this feature enabled, once the policy changes have been made, the ISP can simply do a reconfiguration of the peering without having to tear it down. To support soft-reconfiguration on a peering, the router requires one extra configuration command, for example:

```
router bgp 200
  neighbor 215.17.3.1 remote-as 210
  neighbor 215.17.3.1 soft-reconfiguration in
  neighbor 215.17.3.1 route-map in-filter in
  neighbor 215.17.3.1 route-map out-filter out
```

If the policy on the peering requires to be changed, the ISP makes the changes to the route-map configuration in the example above, and then simply issues the command “*clear ip bgp neighbor 215.17.3.1 soft*”. If only the inbound or outbound policy needs to be changed, the clear command can be supplemented with *in* or *out* directives.

Notice the above configuration. Soft-reconfiguration is only required inbound – outbound does not require to be explicitly configured because to make outbound policy configuration changes the BGP process simply has to send an incremental update to its neighbour.

One caveat, which shouldn't be an issue for ISPs who have read the previous section regarding router memory requirements... Soft-reconfiguration requires more router memory as the router has to store prefixes it has received prior to the BGP inbound policy being implemented. If the inbound policy is complex and/or there are multiple peerings it is possible that twice the amount of memory will be required by the BGP process than if soft reconfiguration was not configured.

It is considered best current practice for ISPs who have external BGP peerings to use soft configuration. The impact on their networks, their customers' perceived service quality, and that of the Internet is too great without this feature. Consider the impact and effects of the following section discussing BGP Route Flap Dampening.

Next-Hop-Self

The BGP command `next-hop-self` simply does as it suggests. The next-hop on BGP announcements from this router are set to the local router-id rather than the IP address of the origin of the prefix. Two common implementation examples are given in the following sections. To configure `next-hop-self`, use the following BGP configuration:

```
router bgp 200
  neighbor 215.17.3.1 remote-as 210
  neighbor 215.17.3.1 next-hop-self
```

External connections

If a prefix is heard from an external network, the next hop is preserved throughout the IGP. However, setting `next-hop-self` on the border router which is distributing the eBGP prefix into iBGP means that the external prefix will receive the router-id (loopback interface IP address) of the border router rather than the external next hop. This feature is especially used by ISPs at exchange points to ensure consistency and reliability of connections across the exchange.

Aggregation routers

If a prefix is injected into the iBGP at a gateway router (the standard way of injecting a customer prefix into the iBGP), the next hop address is the IP address of the point-to-point link between the gateway aggregation router and the customer. This will mean that the iBGP has a large number of next-hop addresses to resolve from the IGP (not bad in itself) and that the IGP will be larger, resulting in slower convergence and greater potential instability in case of instability or failures in the network.

The obvious solution to this is to use `ip unnumbered` – the next-hop address in that case will be the router-id. However, many ISPs who started off using IP addresses on point to point links have a lot of work to do to migrate to IP unnumbered on all their customer point-to-point connections. A quicker fix, which will let them scale the IGP but not require undue haste (i.e. panic!) at renumbering large numbers of customer interfaces, is to set `next-hop-self` on the iBGP peers of the gateway router. The customer prefixes will now appear in the iBGP with the loopback address of the aggregation router, simplifying the route lookups, and more importantly meaning that the IGP no longer has to carry all the point-to-point link addresses used for customer connections.

BGP Dampening

Route flap dampening (introduced in Cisco Internetwork Operating System [Cisco IOS] Release 11.0) is a mechanism for minimising the instability caused by route flapping. Route flapping is the BGP network prefixes being frequently added and removed. Whenever a network goes down, the rest of the Internet would like to know about it. Hence, BGP propagates that state change throughout the Internet. Yet, if this state change is happening from a faulty circuits (frequently going up and down) or from mis-configured routing (redistributed the IGP into the EGP), the Internet would experience several hundred BGP state changes a second. For every state change, BGP must allocate time to work to process the work and pass on the changed to all other BGP neighbours. This places a tremendous strain on the backbone routers. Hence, the tool to control and minimise the effect of route flaps – BGP Dampening.

The following are the commands used to control route dampening:

```
bgp dampening [[route-map map-name] [half-life-time reuse-value suppress-value maximum-suppress-time]]
```

half-life-time – range is 1 – 45 minutes; current default is 15 minutes.

reuse-value – range is 1 – 20000; default is 750.

suppress-value – range is 1 – 20000; default is 2000.

max-suppress-time – maximum duration a route can be suppressed. Range is 1 – 255; default is four times half-life time.

Saturday, February 19, 2000

`show ip bgp dampened-routes` – Display all the damped routes with the time remaining to unsuppress. Very useful for find out which sites are having instability problems.

`clear ip bgp dampening [<address> <mask>]` – Clear the dampening related information. This will also unsuppress the suppressed routes. Very useful when one of your customers call you about a “unreachable” network that has been suppressed.

A route map can be associated with bgp dampening to selectively apply the dampening parameters if certain criteria are found. Example selective dampening criteria include matching on:

- A specific IP route
- AS Path
- BGP community

Adjusting the dampening timers becomes essential when administrators cannot afford to have a long outage for a specific route. BGP dampening with route maps is a powerful tool to selectively penalize ill-behaved routes in a user-configurable and controlled manner.

Detailed examples of BGP Dampening Techniques with route maps can be found in the book *Internet Routing Architectures*, by Bassam Halabi.

Recommended route flap dampening parameters for use by ISPs were composed into a document by the RIPE Routing Working Group (routing-wg@ripe.net) and are available at <http://www.ripe.net/docs/ripe-178.html>. These values are used by many European and US ISPs, and are based on the operational experienced gained in the industry.

How it works. For each flap a penalty (1000) is accessed on the route. As soon as the penalty exceeds the ‘suppress-limit’, the advertisement of route will be suppressed. The penalty will be exponentially decayed based on a pre-configured half-life time. Once the penalty decreases below the *reuse-limit*, it will be unsuppressed.

The routes external to an AS learned via iBGP will not be dampened. This is to avoid the iBGP peers having higher penalty for routes external to the AS. Thus, any route that flaps at a rate more than the half-life time will be eventually suppressed. The penalty will be decayed at a granularity of 5 seconds and the entries will be unsuppressed with the granularity of 10 second. Further, when the penalty is less than half of the reuse-limit, we will purge the damping information.

BGP Flap Statistics: It is possible to monitor the flaps of all the paths that are flapping. The statistics will be lost once the route is not suppressed and stable for at least one half-life time. The display will look like the following:

```
cerdiwen#sh ip bgp neighbors 171.69.232.56 flap-statistics
BGP table version is 18, local router ID is 172.19.82.53
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          From           Flaps Duration Sup-time Path
*> 5.0.0.0          171.69.232.56  1     0:02:21    300
*> 6.0.0.0          171.69.232.56  2     0:03:21    300
```

The following are the new commands that will display flap statistics:

`show ip bgp flap-statistics` – Displays flap statistics for all the paths.

`show ip bgp flap-statistics regexp <regexp>` – Display flap statistics for all paths that match the regular expression

`show ip bgp flap-stastistics filter-list <list>` – Display flap statistics for all paths that pass the filter

`show ip bgp flap-statistics x.x.x.x <m.m.m.m>` – Display flap statistics for a single entry

`show ip bgp flap-statistics x.x.x.x m.m.m.m longer-prefix` – Display flap statistics for more specific entries

`show ip bgp neighbor x.x.x.x flap-statistics` – Display flap statistics for all paths from a neighbor

NOTE: As we maintain information about only one path for a neighbour, the `show ip bgp flap-statistics neighbor` could show a different path for the same NLRI¹⁶.

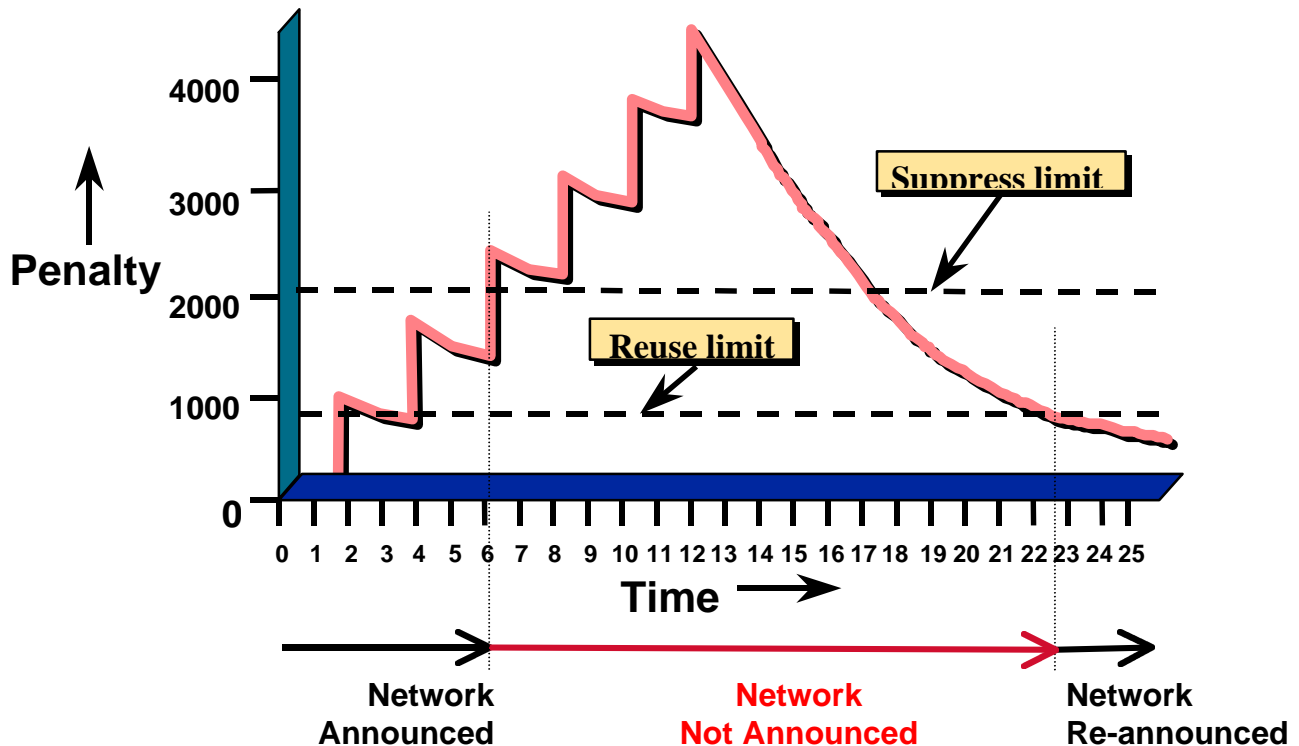


Figure 13 – BGP Route Flap Dampening

The following commands could be used to clear the flap statistics.

`clear ip bgp flap-statistics` – Clear flap statistics for all routes

`clear ip bgp flap-statistics regexp <reg>` – Clear flap statistics for all the paths that match the regular expression

`clear ip bgp flap-statistics filter-list <list>` – Clear flap statistics for all the paths that pass the filter

`clear ip bgp flap-statistics x.x.x.x <m.m.m.m>` – Clear flap statistics for a single entry

`clear ip bgp x.x.x.x flap-statistics` – Clear flap statistic for all paths from a neighbour

BGP Neighbour Authentication

¹⁶ NLRI stands for Network Layer Reachability Information.

Saturday, February 19, 2000

You can invoke MD5 authentication between two BGP peers. This feature must be configured with the same password on both BGP peers; otherwise, the connection between them will not be made. The authentication feature uses the MD5 algorithm. Invoking authentication causes the Cisco IOS software to generate and check the MD5 digest of every segment sent on the TCP connection. If authentication is invoked and a segment fails authentication, then a message appears on the console.

Configuring a password for a neighbour will cause an existing session to be torn down and a new one established. If you specify a BGP peer group by using the *peer-group-name* argument, all the members of the peer group will inherit the characteristic configured with this command. If a router has a password configured for a neighbour, but the neighbour router does not, a message such as the following will appear on the console while the routers attempt to establish a BGP session between them:

```
%TCP-6-BADAUTH: No MD5 digest from [peer's IP address]:11003 to [local router's IP address]:179
```

Similarly, if the two routers have different passwords configured, a message such as the following will appear on the console:

```
%TCP-6-BADAUTH: Invalid MD5 digest from [peer's IP address]:11004 to [local router's IP address]:179
```

The following example specifies that the router and its BGP peer at 145.2.2.2 invoke MD5 authentication on the TCP connection between them:

```
router bgp 109
 neighbor 145.2.2.2 password v61ne0qkel133&
```

Removing Private ASes

Some ISPs use private ASes within their network (typically but not exclusively for customers who multihomed onto their backbone). There is a BGP option which prevents any private ASes from being leaked to the Internet:

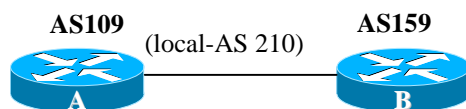
```
router bgp 109
 neighbor 145.2.2.2 remote-private-AS
```

If the BGP update has private ASes in the AS path, this option will remove the private AS numbers. Note that it will not work if there are private and public ASes in the AS path – so this option cannot be used in the case where a network is using a private AS for its BGP, yet providing transit to the public Internet. Note that in the case of BGP confederations, the private AS will be removed so long as the private AS appears after (and outside) the confederation portion of the AS path. Finally, this command can only be used for eBGP peers – it has no effect for iBGP peers.

BGP local-as

Configuration

A recent addition to BGP is the **local-as** neighbour option, which allows the AS number of the network to be changed for eBGP peerings. It isn't possible in IOS to configure BGP to run in more than one AS – the local AS feature provides a “solution” to this “problem”.



Router A is in AS 109, and Router B is in AS 159. However, when A peers with B it uses AS 210 as its AS number. As far as B is concerned it is peering with a router in AS 210. Operationally this is equivalent to Router B peering with a

router in AS 210, and that router peering with Router A in AS 109. The AS path for all prefixes learned from Router B as seen on Router A would be 210_159 – AS 210 is inserted into the AS path sequence. The AS path for all prefixes learned from Router A by Router B would be 210_109.

Router A configuration:

```
router bgp 109
  neighbor 145.2.2.2 remote-as 159
  neighbor 145.2.2.2 local-as 210
```

Router B configuration:

```
router bgp 159
  neighbor 144.2.2.1 remote-as 210
```

Local-as can be configured per eBGP peer, or per peer-group. It cannot be configured for individual members of a peer-group. Also, local-as cannot be configured with the AS number of the BGP process of the local router, nor can it be configured with the AS number of the BGP process on the remote router. **Local-as** cannot be used between two confederation eBGP peers – they must be true eBGP peers.

Motivation

The local-as feature is most often used and was requested by ISPs who are active in the acquisition trail. Basically if an ISP purchases a smaller ISP, there is a significant amount of working which goes into renumbering the smaller ISPs backbone into the same AS number as that of the purchaser. However, there are situations where the smaller ISP may have agreements with peers, or a specific eBGP configuration which is hard to migrate to the new AS (could be political or technical reasons, or both). The local-as option can make the peering router in AS109 look as though it really is in AS210, the ISP network which was purchased, until the political issues with the inter-provider peering can be sorted out.

BGP Neighbour Changes

It is possible to log bgp neighbour state changes to a Unix syslog server. This is extremely useful for most syslog based monitoring systems as it gives early warning of problems with iBGP peers, and more especially external BGP neighbours. The logging is enabled by:

```
router bgp 109
  bgp log-neighbor-changes
```

Limiting the Number of Prefixes from a Neighbour

There have been times, either via configuration error or a blatant attack on the Internet, that the Global *Default Free* Routing Table jumped to 2 to 3 times its size. This has caused severe problems on sections of the Internet. The BGP neighbour command `maximum-prefix` was added to help networks safe guard against these sorts of problems.

This command allows you to configure a maximum number of prefixes a BGP router is allowed to receive from a peer. It adds another mechanism (in addition to distribute lists, filter lists, and route maps) to control prefixes received from a peer. When the number of received prefixes exceeds the *maximum* number configured, the router terminates the peering (by default). However, if the keyword **warning-only** is configured, the router instead only sends a log message, but continues peering with the sender. If the peer is terminated, the peer stays down until the **clear ip bgp** command is issued.

In the following example, the maximum number of prefixes allowed from the neighbour at 129.140.6.6 is set to 100000 (the Global Internet Route Table was around 71000 at the time of writing):

```
router bgp 109
  network 131.108.0.0
  neighbor 129.140.6.6 maximum-prefix 100000
```

Saturday, February 19, 2000

The maximum-prefix command sends log messages also, so any overrun can be trapped by a management system which monitors the router's syslog output. One message is sent when the number of prefixes received reaches the configured threshold value:

```
%BGP-4-MAXPFX: No. of unicast prefix received from 129.140.6.6 reaches 78351, max 100000
```

The default threshold is 75%. This can be changed by specifying the threshold percentage in the maximum-prefix line – the following example sets the threshold to 95%:

```
router bgp 109
  bgp log-neighbor-changes
  neighbor 129.140.6.6 maximum-prefix 100000 95
```

Another message is sent when the number of prefixes received exceed the maximum number of prefixes configured. Logging of neighbour changes is included for completeness in this example:

```
%BGP-3-MAXPFXEXCEED: No. of unicast prefix received from 129.140.6.6: 103411 exceed limit 100000
%BGP-5-ADJCHANGE: neighbor 129.140.6.6 Down - BGP Notification Sent
```

BGP Fast External Fallover

By default if a BGP peer doesn't respond within a few seconds, the peering relationship will be reset. By adding the *no bgp fast-external-fallover* configuration, the peering will be held open for considerably longer. This configuration is desirable, if not essential, in the case of long distance peering links, or unreliable or long latency connections to other AS's, and in the case where ISPs prefer stability over convergence speed in large networks.

```
router bgp 109
  no bgp fast-external-fallover
```

Important note: this configuration option should be used with care. It is recommended that fast-external-fallover is only used for links to an ISP's upstream provider, or over unreliable links. Because the fallover is slower, it is possible to blackhole routes for up to 3 minutes. This may prove problematic on, for example, a link to a multihomed customer, where their peering may have suffered an unintentional reset due to human activity.

BGP Peer-group¹⁷

Summary

The major benefits of BGP peer-groups are the reduction of resource (CPU load and memory) required in update generation. Another benefit is that it simplifies BGP configuration.

With BGP peer-groups, the routing table is walked only once and updates are replicated to all other peer-group members that are in sync. Depending on the number of members, the number of prefixes in the table and the number of prefixes advertised, this could significantly reduce the load. It is thus highly recommended that peers with identical outbound announcement policies be grouped into peer-groups.

Requirements

All members of a peer-group must share identical outbound announcement policies (e.g., distribute-list, filter-list, and route-map), except for the originating default which is handled on a per-peer basis even for peer-group members.

The inbound update policy can be customised for each individual member of a peer-group.

¹⁷ Thanks to Enke Chen for providing most of the text describing BGP Peer Groups.

A peer-group must be either internal (with iBGP members) or external (with eBGP members). Members of an external peer-group have different AS numbers.

Historical Limitations

There used to be several limitations with BGP peer-groups:

- If used for clients of a route reflector, all the clients should be fully meshed.
- If used as eBGP peer-group, transit can not be provided among the peer-group members.
- All the eBGP peer-group members should be from the same subnet to avoid non-connected nexthop announcements.

Inconsistent routing would occur if these limitations were not followed. They have been removed starting with the following IOS versions: 11.1(18)CC, 11.3(4), and 12.0. Only the router on which the peer-groups are defined needs to be upgraded to the new code.

Typical Peer-group Usage

Typically, ISP Network Engineers group BGP peers on a router into peer-groups based on their outbound update policies. A list of peer-groups commonly by ISPs are listed as follows:

- Normal iBGP peer-group: for normal iBGP peers.
- iBGP Client peer-group: for reflection peers on a route reflector.
- eBGP Full-routes: for peers to receive full Internet routes.
- eBGP customer-routes: for peers to receive routes from direct customers of the ISP only. Some members can be configured with “default-origination” to receive the default route as well as the customer routes.
- eBGP default-routes: for peers to receive the default route, and possibly along with a few other routes.

BGP Peer-Group Examples

This example shows an iBGP Peer-Group for a router inside an ISP’s backbone:

```
router bgp 109
neighbor internal peer-group
neighbor internal remote-as 109
neighbor internal update-source loopback 0
neighbor internal send-community
neighbor internal route-map send-domestic out
neighbor internal filter-list 1 out
neighbor 131.108.10.1 peer-group internal
neighbor 131.108.20.1 peer-group internal
neighbor 131.108.30.1 peer-group internal
neighbor 131.108.30.1 filter-list 3 in
```

This example shows an eBGP Peer-Group for a router peering with several ISPs all with the same advertisement policies:

```
router bgp 109
neighbor external-peer peer-group
neighbor external send-community
neighbor external-peer route-map set-metric out
neighbor external-peer route-map filter-peer in
neighbor 160.89.1.2 remote-as 200
neighbor 160.89.1.2 peer-group external-peer
neighbor 160.89.1.4 remote-as 300
neighbor 160.89.1.4 peer-group external-peer
```

Saturday, February 19, 2000

Using Prefix-list in Route Filtering¹⁸

Introduction

The prefix-list feature offers significant performance improvement (in terms of CPU consumed) over the access-list in route filtering of routing protocols. It also provides for faster loading of large lists, and support for incremental configuration. In addition, the command line interface is much more intuitive. This feature is available in IOS versions from 11.1CC(17), 11.3(3) and 12.0.

The prefix-list preserves several key features of access-list:

- Configuration of either “permit” or “deny”.
- Order dependency – first match wins.
- Filtering on prefix length – both exact match and range match.

However, prefix-lists, or prefix-lists in route-maps does not support packet filtering. This documents presents the detailed configuration commands and several applications of the prefix-list in route filtering.

Configuration Commands

There are three configuration commands related to the prefix-list.

```
no ip prefix-list <list-name>
```

where <list-name> is the string identifier of a prefix-list. This command can be used to delete (i.e., destroy) a prefix-list.

```
[no] ip prefix-list <list-name> description <text>
```

This command can be used to add/delete a text description for a prefix-list.

```
[no] ip prefix-list <list-name> [seq <seq-value>] deny|permit \ <network>/<len> [ge <ge-value>]  
[le <le-value>]
```

This command can be used to configure or delete an entry of a prefix-list.

Command Attributes

<list-name>: Mandatory. A string identifier of a prefix-list.

seq <seq-value>: Optional. It can be used to specify the sequence number of an entry of a prefix list. By default, the entries of a prefix list would have sequence values of 5, 10, 15 and so on. In the absence of a specified sequence value, the entry would be assigned with a sequence number of (Current_Max+ 5).

Like access-list, a prefix-list is an ordered list. The number is significant when a given prefix matches multiple entries of a prefix list, the one with the smallest sequence number is considered as the real match.

deny|permit: Mandatory. An action taken once a match is found.

<network>/<len>: Mandatory. The prefix (i.e., network and prefix length). Multiple policies (exact match or range match) with different sequence numbers can be configured for the same <network>/<len>.

ge <ge-value>: Optional.

¹⁸ The core of this section is by Bruce R. Babcock [bbabcock@cisco.com] and Enke Chen

le <le-value>: Optional.

Both “ge” and “le” are optional. They can be used to specify the range of the prefix length to be matched for prefixes that are more specific than <network>/<len>. Exact match is assumed when neither “ge” nor “le” is specified. The range is assumed to be from “ge-value” to 32 if only the “ge” attribute is specified. And the range is assumed to be from “len” to “le-value” if only the “le” attribute is specified.

A specified <ge-value> and/or <le-value> must satisfy the following condition:

```
len < ge-value < le-value <= 32
```

Configuration Examples

SPECIFICATION OF EXACT PREFIXES

Deny the default route 0.0.0.0/0	<code>ip prefix-list abc deny 0.0.0.0/0</code>
Permit the prefix 35.0.0.0/8	<code>ip prefix-list abc permit 35.0.0.0/8</code>

SPECIFICATION OF GROUP OF PREFIXES

In 192/8, accept up to /24	<code>ip prefix-list abc permit 192.0.0.0/8 le 24</code>
In 192/8, deny /25+	<code>ip prefix-list abc deny 192.0.0.0/8 ge 25</code>
In all address space, permit /8 – /24	<code>ip prefix-list abc permit 0.0.0.0/0 ge 8 le 24</code>
In all address space, deny /25+	<code>ip prefix-list abc deny 0.0.0.0/0 ge 25</code>
In 10/8, deny all	<code>ip prefix-list abc deny 10.0.0.0/8 le 32</code>
In 204.70.1/24, deny /25+	<code>ip prefix-list abc deny 204.70.1.0/24 ge 25</code>
Permit all	<code>ip prefix-list abc permit 0.0.0.0/0 le 32</code>

INCREMENTAL CONFIGURATION

A prefix-list can be re-configured incrementally, that is, an entry can be deleted or added individually. For example, to change a prefix-list from the initial configuration to a new configuration, only the difference between the two needs to be deployed as follows:

The initial configuration:

```
ip prefix-list abc deny 0.0.0.0/0 le 7
ip prefix-list abc deny 0.0.0.0/0 ge 25
ip prefix-list abc permit 35.0.0.0/8
ip prefix-list abc permit 204.70.0.0/15
```

The new configuration:

```
ip prefix-list abc deny 0.0.0.0/0 le 7
ip prefix-list abc deny 0.0.0.0/0 ge 25
ip prefix-list abc permit 35.0.0.0/8
ip prefix-list abc permit 198.0.0.0/8
```

The difference between the two configurations:

Saturday, February 19, 2000

```
no ip prefix-list abc permit 204.70.0.0/15
ip prefix-list abc permit 198.0.0.0/8
```

SPECIAL NOTE ON THE SEQUENCE NUMBER

The sequence number is used internally to identify the “real” match (the one with the lowest sequence number) when multiple prefix-list entries match a given prefix. It can also be used to insert an entry to a specific relative position (e.g., sequence number of 7). However, in most cases a prefix-list can be structured such that there is no need to specify sequence numbers, and such an approach would make it easier to automate prefix-list generation, “diff” generation, and deployment. The sequence numbers can be “switched off” from appearing in the configuration by the command:

```
no ip prefix-list sequence-number
```

How Does Match Work

The matching is similar to that of the access-list. More specifically:

- An empty prefix-list would permit all prefixes.
- An implicit deny is assumed if a given prefix does not match any entries of a prefix-list.
- When multiple entries of a prefix-list match a given prefix, the one with the smallest sequence is considered as the “real” match. In short, the first match wins!

Here is an example to illustrate the “first match rule”. Supposed a prefix-list is configured as follows:

```
ip prefix-list abc deny 10.0.0.0/8 le 32
ip prefix-list abc permit 0.0.0.0/0 le 32
```

Then, a given prefix 10.1.0.0/16 would match both entries. However, the prefix will be “denied”, as the first entry is the real match.

Show and Clear Commands

`show ip prefix-list [detail|summary]` – Displays information of all prefix-lists.

`show ip prefix-list [detail|summary] [<name>]` – Displays information of a prefix-list.

`show ip prefix-list <name> [seq <seq-num>]` – Display the prefix-list entry with the given sequence number

`show ip prefix-list <name> <network>/<len>` – Displays the policy associated with the node <network>/<len>

`show ip prefix-list <name> <network>/<len> longer` – Displays all entries of a prefix list that are more specific than the given <network>/<len>

`show ip prefix-list <name> <network>/<len> first-match` – Displays the entry of a prefix list that matches the given <network>/<len>

`clear ip prefix-list [<name>] [<network>/<len>]` – Resets the “hit count” of prefix-list entries

Using Prefix-list with BGP

The prefix-list can be used as an alternative to the BGP `neighbor x.x.x.x distribute-list` command. The configuration of prefix-list and distribute-list for a BGP peer are mutually exclusive.

```
router bgp xxx
neighbor x.x.x.x prefix-list <name> in|out
```


Using Prefix-list in Route-map

The prefix-list can be used as an alternative to access-lists used in the command `match ip address|next-hop|route-source <access-list>` of a route-map. The configuration of prefix-lists and access-lists are mutually exclusive within the same sequence of a route-map.

```
route-map <name> permit|deny <seq-num>
  match ip address|next-hop|route-source prefix-list <name> [<name> ...]
```

Besides its application in BGP, route-maps using prefix-lists can be used for route filtering, default-origination, and redistribution in other routing protocols as well. For example, the following configuration can be used to conditionally originate a default route (0.0.0.0/0) when there exists a prefix 10.1.1.0/24 in the routing table:

```
ip prefix-list cond permit 10.1.1.0/24
!
route-map default-condition permit 10
  match ip address prefix-list cond
!
router rip
  default-information originate route-map default-condition
```

Using Prefix-list in Other Routing Protocols

The prefix-list can be used to filter inbound and outbound routing updates, as well to control route redistribution between different routing protocols. Compared with using the access-list, prefix-list based filtering offers the ability of prefix length filtering. It also has the flexibility of filtering either the prefix, or the gateway, or both for incoming updates.

As usual, access-list and prefix-list are mutually exclusive in one “distribute-list” command.

FILTERING ON INBOUND UPDATES

Inbound updates can be filtering on the prefix, or the gateway or both prefix and gateway:

```
router rip | igmp | eigrp
  distribute-list {prefix <name1>} | {gateway <name2>} | {prefix <name1> gateway <name2>} in
[<interface>]
```

Where <names> is the name of a prefix-list to be applied to the prefix being updated, and <name2> the name of a prefix-list to be applied to the gateway (i.e., next-hop) of a prefix being updated. The filtering can also be specified with a specific interface.

FILTERING ON OUTBOUND UPDATES

```
router rip | igmp | eigrp ...
  distribute-list prefix <name1> out [<routing_process> | <interface>]
```

EXAMPLE

In the following configuration, the RIP process will only accept prefixes with prefix length of /8 to /24:

```
router rip
  version 2
  network x.x.x.x
  distribute-list prefix max24 in
!
ip prefix-list max24 seq 5 permit 0.0.0.0/0 ge 8 le 24
```

Also, the following configuration will make RIP accept routing update only from 192.1.1.1, besides the filtering on prefix length:

```
router rip
  distribute-list prefix max24 gateway allowlist in
  !
  ip prefix-list allowlist seq 5 permit 192.1.1.1/32
```

BGP Conditional Advertisement

Conditional advertisement of prefixes has been introduced into 11.1CC, 11.2, 12.0 and more recent versions of IOS in an effort to contribute to the stability of large BGP based networks (specifically the Internet). Conditional advertisement is usually configured when an AS has at least two connections to another AS. The inter-AS peering routers “watch” the links between the ASes – if one link fails, prefixes are advertised out of the other link. This allows ISPs to set up efficient and effective multihoming without leaking many subprefixes to the Internet, yet retain a good degree of backup in the case of uplink failure.

The configuration command is this:

```
router bgp 109
  neighbor 129.140.6.6 remote-as 159
  neighbor 129.140.6.6 advertise-map announce non-exist-map monitor
```

The **non-exist-map** describes the prefix which will be monitored by the BGP router. The **advertise-map** statement describes the prefix which will be advertised when the prefix in non-exist-map has disappeared from the bgp table. The route-maps **announce** and **monitor** are standard IOS route-maps and are used to configure the prefixes which will be part of the conditional advertisement process.

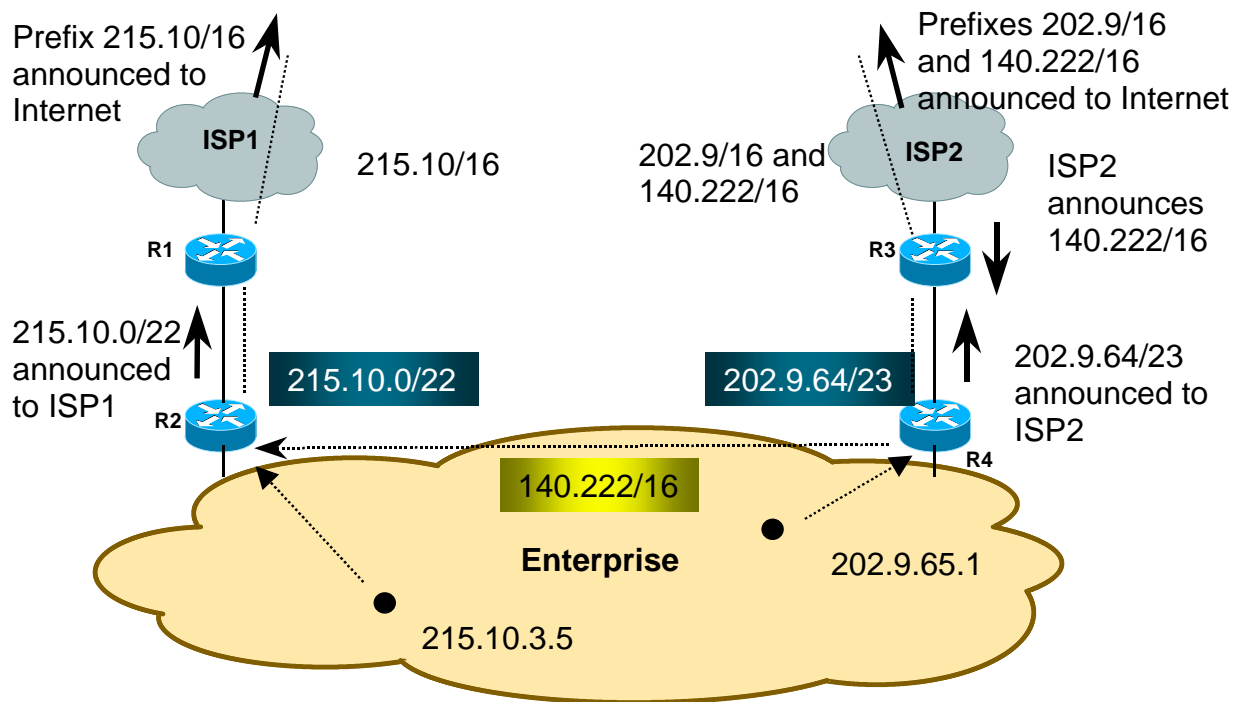


Figure 14 – Conditional Advertisement – Steady State

Example

Consider the example depicted in Figure 14. This shows a dual homed enterprise network (AS300) which has received address space from its two upstream ISPs. It announces the 215.10.0.0/22 prefix to ISP1 (AS100), and the 202.9.64/23

prefix to ISP2 (AS200). These networks are part of the respective upstream ISPs address blocks, so all the Internet will see are the two aggregates as originated by ISP1 and ISP2. This is the steady state situation.

So that conditional advertisement can be used, ISP2 needs to announce a prefix to the Enterprise. In this example, it uses its own 140.222/16 address block – the Enterprise needs run iBGP between its two border routers, basically so that R2 has the 140.222/16 prefix in its BGP table. (Note, it doesn't have to be ISP2's /16 block – it can be any prefix agreed between the Enterprise and their upstream, so long as a prefix is announced.)

Consider now the case where the link from the Enterprise to ISP2 fails, for some reason. Because the link fails, the Enterprise router R4 no longer hears the 140.222/16 announcement from ISP2. R4's iBGP session with R2 no longer advertises this prefix, so R2 no longer hears 140.222/16. This is the required condition to activate the conditional advertisement – 140.222/16 is no longer in R2's BGP table. R2 now starts advertising the 202.9.64/23 prefix to ISP1 so that connectivity to that prefix is maintained during the failure of the link to ISP2. The example after link failure is shown in Figure 15.

The configuration to achieve this is only required on Router 2, and is as follows:

```

! Router 2 configuration
router bgp 300
  neighbor <R1> remote-as 100
  neighbor <R1> advertise-map ISP2-subblock non-exist-map ISP2-backbone
!
route-map ISP2-subblock permit 10
  match ip address 1
!
route-map ISP2-backbone permit 10
  match ip address 2
!
access-list 1 permit 202.9.64.0 0.0.1.255
access-list 2 permit 140.222.0.0 0.0.255.255
!

```

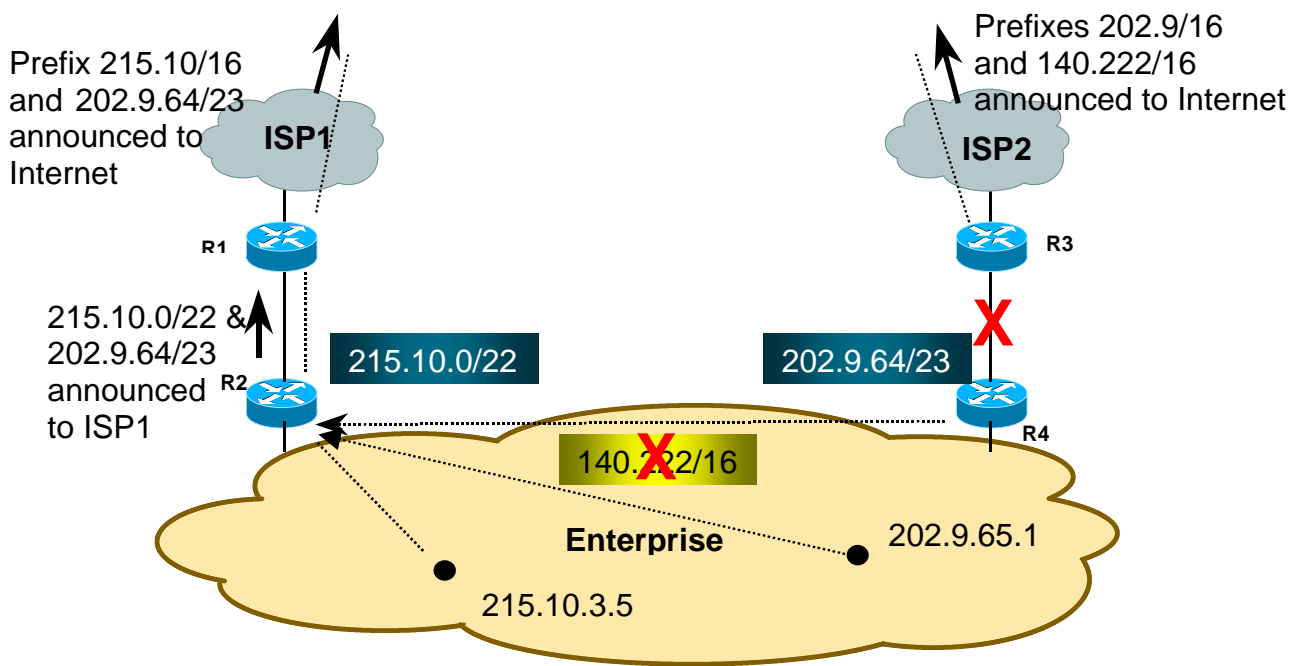


Figure 15 – Conditional Advertisement – Failure Mode

Saturday, February 19, 2000

Access-list 1 is the prefix which will be announced during failure mode. Access-list 2 describes the prefix which is monitored to detect whether a failure has happened.

Router4 can, ofcourse, be configured with a similar conditional advertisement statement, monitoring a prefix announced by ISP1 towards Router2.

Summary: conditional advertisements ensure that a multihomed AS will have good backup/failure modes without having to leak subprefixes into the Internet routing table unnecessarily. Only in the case of link failure will a sub prefix be leaked – consider the effects of route flap dampening using conditional advertisements, as opposed to using standard BGP backup configuration.

BGP Route Refresh

A new feature available from IOS 12.0(5)S is route refresh. The concept is similar to soft reconfiguration but is a capability shared between two BGP speakers (as opposed to soft reconfiguration which is configured on the local router only).

If the local router requires the a fresh view of the routing table it can send a “route refresh” request to the neighbouring BGP peer. This would be required, for example, when the inbound routing policy has been changed. On receipt of the route refresh request, the remote router would send its list of prefixes to the requesting router.

Route refresh capability requires no extra memory on the local router. Where the capability exists between speakers, it is recommended that this is chosen over soft reconfiguration (as the latter requires more memory to store the inbound prefixes received from the remote peer). To request a route refresh, use the command:

```
clear ip bgp <neighbour> soft in
```

No other configuration is required.

BGP Outbound Route Filter Capability

This new feature, supported from IOS 12.0(5)S onwards, allows one BGP speaker to install its inbound locally configured prefix-list filter on to the remote BGP speaking router. This is especially used for reducing the amount of unwanted routing update from the remote peer.

The remote BGP speaker would apply the received prefix-list filter, in addition to its locally configured outbound filters (if any), to constrain/filter its outbound routing updates to the neighbor. This mechanism can be used to avoid unwanted routing updates and thus help reduce resources required for routing update generation and processing.

For example, Prefix-List ORF can be used to address the issue of receiving (“unwanted”) full routes from multihomed BGP customers. The customer can simply enable this feature on their router and thus allow their providers to manage the filtering of their route announcements. This avoids unwanted routing updates coming from the customer to their upstream ISP.

Currently the Prefix-List ORF is implemented for IPv4 unicast only. Some points to note about the implementation:

- By default, the Prefix-List ORF Capability is not advertised to any neighbours
- The capability can not be advertised to a neighbour that is a peergroup member.
- The Prefix-List ORF is pushed over to the peer router immediately after the session is established if the local router has received the ORF capability, and has configured inbound prefix-list filter for the neighbour.

Configuration

The router configuration command is included in the following example:

```
router bgp Y
neighbor x.x.x.x remote-as Z
neighbor x.x.x.x description Peer router R2
neighbor x.x.x.x capability prefix-filter
neighbor x.x.x.x prefix-list FilterZ-in in
```

This command can be used to enable the advertisement of the Prefix-List ORF Capability to a neighbour. Using the “no neighbor x.x.x.x capability prefix-filter” command disables the Prefix-List ORF Capability.

When the BGP peering is established in this example, the above router (R1) will push its prefix-list “FilterZ-in” over to its peer router x.x.x.x (R2). R2 will receive the prefix-list filter and apply it to its outbound update to R1 (in addition to its local policy, if any is configured).

Pushing out a Prefix-list ORF

The command to push out a Prefix-list ORF and receive route refresh from a neighbour is:

```
clear ip bgp x.x.x.x in prefix-filter
```

When the inbound prefix-list changes (or is removed), this command can be used to push out the new prefix-list, and consequently receive route refresh from the neighbour based on the new prefix-list. The keyword "prefix-filter" will be ignored if the Prefix-list ORF Capability has not been received from the neighbour.

Without the keyword “prefix-filter”, the command:

```
clear ip bgp x.x.x.x in
```

would simply perform the normal route refresh from the neighbour. It does not push out the current inbound prefix-list filter to the neighbour. The command is useful when inbound routing policies other than the prefix-list filter such as route-map changes.

Displaying Prefix-list ORF

The command to display the prefix-list ORF received from a neighbour is:

```
show ip bgp neighbor x.x.x.x received prefix-filter
```

This will display the received prefix-list. Changes to the output of “show ip bgp neighbor x.x.x.x” are:

```
Prefixlist ORF
Capability advertised; received
Filter sent; received (25 entries)
```

BGP Policy Accounting

Overview

BGP Policy Accounting allows you to account for IP traffic differentially by assigning counters based on community-list, AS number, and/or AS-path on a per input interfaces basis.

Using BGP Policy Accounting, you can account for traffic (and apply billing) according to the route specific traffic traverses. This way, eg. domestic, international, terrestrial, satellite, etc. traffic can all be identified and accounted for on a per customer basis.

This feature takes advantage of BGP's table_map ability to classify the prefixes that it puts into the routing table according to community-lists, AS-path, AS number, etc. Based on those match criteria this feature will set a bucket number (currently 1 to 8) of an accounting table that is associated with each interface. Each bucket thus represents a traffic classification.

Saturday, February 19, 2000

This allows IP traffic to be accounted differentially by community-list, AS number, AS-path per input interface.

Configuration

Specify communities into community-lists (or define AS-path lists, etc) which will classify traffic for accounting.

```
ip community-list 30 permit 100:190
ip community-list 40 permit 100:198
ip community-list 50 permit 100:197
ip community-list 60 permit 100:296
ip community-list 70 permit 100:201
!
```

Define a route-map to match community-lists and set appropriate bucket number:

```
route-map set_bucket permit 10
  match community 30
  set traffic-index 2          ! <-- look here
!
route-map set_bucket permit 20
  match community 40
  set traffic-index 3
!
route-map set_bucket permit 30
  match community 50
  set traffic-index 4
!
route-map set_bucket permit 40
  match community 60
  set traffic-index 5
!
route-map set_bucket permit 50
  match community 70
  set traffic-index 6
```

Modify the bucket number when the IP routing table is updated with BGP learned routes:

```
router bgp 110
  table-map set_bucket        ! <-- look here
  network 15.1.1.0 mask 255.255.255.0
  neighbor 14.1.1.1 remote-as 100
!

ip classless
ip bgp-community new-format
```

Enable the policy accounting feature on the input-interface connected to customer:

```
interface POS7/0
  ip address 15.1.1.2 255.255.255.252
  no ip directed-broadcast
  bgp-policy accounting      ! <-- Look here
  no keepalive
  crc 32
  clock source internal
```

Each customer is off of an input interface (swidb, so can be subinterfaces) which will have the above displayed table of counters associated with it).

Displaying BGP Policy Accounting status

To inspect which prefix is assigned which bucket and which communit[y/ies]

```
Router#sh ip cef 196.240.5.0 detail
```

```

196.240.5.0/24, version 21, cached adjacency to POS7/2
0 packets, 0 bytes, traffic_index 4          ! <-- Look Here
  via 14.1.1.1, 0 dependencies, recursive
    next hop 14.1.1.1, POS7/2 via 14.1.1.0/30
    valid cached adjacency

Router#sh ip bgp 196.240.5.0
BGP routing table entry for 196.240.5.0/24, version 2
Paths: (1 available, best #1)
  Not advertised to any peer
  100
    14.1.1.1 from 14.1.1.1 (32.32.32.32)
      Origin IGP, metric 0, localpref 100, valid, external, best
      Community: 100:197          ! <-- Look Here

```

To look at traffic statistics per-interface:

```

LC-Slot7#sh cef interface traffic-statistics
:
POS7/0 is up (if_number 8)
Bucket      Packets      Bytes
-----
1           0             0
2           0             0
3           50            5000
4          100           10000
5          100           10000
6           10            1000
7           0             0
8           0             0

```

Displaying BGP Policy Accounting statistics

The statistics are stored in a table of packet/byte counters per input software interface (with the assumption that each customer is connected to an input software interface). You can display them by **show cef interface <int> policy-stat**. SNMP support will be added soon.

The statistics are actually displayed per configured table-map match category. Using a route-map you can “match” against configured community-lists, AS-paths, etc and “set” to correspond to a specific “bucket/index” in the above mentioned table.

FURTHER STUDY AND TECHNICAL REFERENCES

In addition to the references mentioned throughout the text, the following are pointers to technical references available on the Internet that highlights foundation details (e.g. how does OSPF work).

Internet Routing Architectures, New Riders Publishing (Cisco Press). ISBN 1-56205-652-2. Author: Bassam Halabi.

RFC2267 Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing. P. Ferguson, D. Senie, January 1998.

<http://info.internet.isi.edu:80/in-notes/rfc/files/rfc2267.txt>

Using the Border Gateway Protocol for Interdomain Routing, Cisco Connection On-line (CCO) Web site.

<http://www.cisco.com/univercd/cc/td/doc/cisintwk/ics/icsbgp4.htm>

Internetworking Technology Overview. On-line whitepapers and tutorials on the essentials of routing and switching. Cisco Connection On-Line (CCO) Web and the Cisco Documentation CD.

http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/index.htm

Technology Information and Whitepapers. Key references and practical internetworking examples. Cisco Connection On-Line (CCO) Web and the Cisco Documentation CD.

<http://www.cisco.com/univercd/cc/td/doc/cisintwk/index.htm>

APPENDIX 1 – ACCESS LIST AND REGULAR EXPRESSIONS

Access List Types

<1-99>	IP standard access list
<100-199>	IP extended access list
<200-299>	Protocol type-code access list
<700-799>	48-bit MAC address access list
<1100-1199>	Extended 48-bit MAC address access list
<1300-1999>	IP standard access list (expanded range)
<2000-2699>	IP extended access list (expanded range)
compiled	Enable IP access-list compilation (new from 12.0(6)S)
rate-limit	Simple rate-limit specific access-list
permit	Specify packets to forward
deny	Specify packets to reject
dynamic	Specify a DYNAMIC list of PERMITs or DENYs
<0-255>	An IP protocol number
ahp	Authentication Header Protocol
eigrp	Cisco's EIGRP routing protocol
esp	Encapsulation Security Payload
gre	Cisco's GRE tunneling
icmp	Internet Control Message Protocol
igmp	Internet Gateway Message Protocol
igrp	Cisco's IGRP routing protocol
ip	Any Internet Protocol
ipinip	IP in IP tunneling
nos	KA9Q NOS compatible IP over IP tunnelling
ospf	OSPF routing protocol
pcp	Payload Compression Protocol
pim	Protocol Independent Multicast
tcp	Transmission Control Protocol
udp	User Datagram Protocol
a.b.c.d	source or destination address
any	any source host
host	a single source host (equivalent to a.b.c.d 255.255.255.255)
log	log matches against this entry
log-input	log matches against this entry, including input interface
precedence	match packets with given precedence value
tos	match packets with given TOS value

There are other options, depending on which IP protocol has been chosen. For example, TCP has these further options for configuring an access-list:

```
ack, eq, established, fin, gt, log, log-input, lt, neq, precedence, psh,
range, rst, syn, tos, urg
```

Saturday, February 19, 2000

Basic Regular Expressions

These are only some examples of regular expressions. Please refer to the documentation for more in depth discussion and detailed examples.

<code>^200\$</code>	match AS200 only
<code>.*</code>	match all ASes starting with the local AS
<code>^*</code>	match all ASes
<code>^\$</code>	match this AS only
<code>^200_</code>	match all ASes received from AS200
<code>_200_</code>	match all ASes which have AS200 in the path
<code>_200\$</code>	match all ASes with AS200 origin only, whatever the path
<code>^200_210\$</code>	match AS210 origin and received from AS200 only
<code>_200_210_</code>	match all ASes which have been through AS200 \leftrightarrow AS210 link
<code>^(200_)+\$</code>	match AS200, or AS200 with path stuffing ¹⁹
<code>^(_[0-9]+)\$</code>	matches one AS, or one AS with path stuffing by the same AS

¹⁹ AS path stuffing means seeing an AS path such as 200_200_200 for a network announcement. This is commonly used when defining particular routing policy, for example loadsharing.

APPENDIX 2 – CUT AND PASTE TEMPLATES

The following are some cut and paste template you can modify to configure your routers. *Make sure you change the IP addresses and AS numbers* in the templates!

General System Template

```
no service finger
no service pad
no service udp-small-servers
no service tcp-small-servers
no ip bootp server
service nagle
service timestamps debug datetime localtime show-timezone msec
service timestamps log datetime localtime show-timezone msec
service tcp-keepalives-in
no ip source-route
ip spd enable
logging buffered 16384
logging trap debugging
logging x.x.x.x
ip subnet-zero
ip classless
```

General Interface Template

```
no ip redirect
no ip direct broadcast
no ip proxy-arp
no cdp enable
```

General Security Template

```
service password-encryption
enable secret <removed>
no enable password
```

General BGP template

This generic template should be used for configuring BGP on a router. It also includes sample configuration for a peer-group.

```
router bgp 65280
  bgp dampening
  no synchronisation
  no auto-summary
  bgp fast-external-fallover
  bgp log-neighbor-changes
  neighbor ibgp-peer peer-group
  neighbor ibgp-peer remote-as 65280
  neighbor ibgp-peer send-community
  neighbor ibgp-peer update-source loopback 0
```

Saturday, February 19, 2000

Martian and RFC1918 Networks Template

This list represents the common filtering practice of several ISPs. It includes default, multicast, and RFC1918 networks, as well as the so-called Martian networks as can be found in <http://www.ietf.org/internet-drafts/draft-manning-dsua-01.txt>. The use of these filters on inbound and outbound interfaces of border routers is recommended.

Note: the list of these networks is updated and discussed quite frequently by groups such as NANOG (nanog@merit.edu) and IEPG (iepg@iepg.org).

IP Access-List Example

```
! access-list 150 to deny RFC1918 and Martian networks
access-list 150 deny ip 0.0.0.0 0.255.255.255 255.0.0.0 0.255.255.255
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 deny ip any 255.255.255.128 0.0.0.127
access-list 150 permit ip any any
!
```

IP Prefix-List Example

```
ip prefix-list rfc1918-dsua description Networks which shouldn't be announced
ip prefix-list rfc1918-dsua deny 0.0.0.0/8 le 32
ip prefix-list rfc1918-dsua deny 10.0.0.0/8 le 32
ip prefix-list rfc1918-dsua deny 127.0.0.0/8 le 32
ip prefix-list rfc1918-dsua deny 169.254.0.0/16 le 32
ip prefix-list rfc1918-dsua deny 172.16.0.0/12 le 32
ip prefix-list rfc1918-dsua deny 192.0.2.0/24 le 32
ip prefix-list rfc1918-dsua deny 192.168.0.0/16 le 32
ip prefix-list rfc1918-dsua deny 224.0.0.0/3 le 32
ip prefix-list rfc1918-dsua deny 0.0.0.0/0 ge 25
ip prefix-list rfc1918-dsua permit 0.0.0.0/0 le 32
!
```

BGP Flap Dampening Configuration

Recommended route flap dampening parameters for use by ISPs were composed into a document by the RIPE Routing Working Group and are available at <http://www.ripe.net/docs/ripe-178.html> and <http://www.ripe.net/docs/ripe-178.ps> in Postscript form. These values are used by many European and US ISPs, and are based on the operational experience gained in the industry.

The configuration examples are reproduced here for convenience – the values have been updated to include recent changes in the locations of the root nameservers.

IP Access-List Example

This is a configuration example using the IP access-list, similar to what is quoted in the ripe-178 document. Note that the access-list 180 covering the root-nameserver networks has been updated with the most recent values. Check before implementing these filters (you could use the Unix command “`dig . ns`” to find out, for example).

```
router bgp 65280
  bgp dampening route-map RIPE178-flap-dampen
```

```

! no flap dampening for special user defined networks defined in access-list 183
route-map RIPE178-flap-dampen deny 10
  match ip address 183
! no flap dampening for root nameserver networks in access-list 180
route-map RIPE178-flap-dampen deny 20
  match ip address 180
! flap dampening for all the /24 and longer prefixes
route-map RIPE178-flap-dampen permit 30
  match ip address 181
  set dampening 30 750 3000 60
! flap dampening for all /22 and /23 prefixes
route-map RIPE178-flap-dampen permit 40
  match ip address 182
  set dampening 15 750 3000 45
! flap dampening for all remaining prefixes
route-map RIPE178-flap-dampen permit 50
  set dampening 10 1500 3000 30

! Access Lists for route flap dampening as per RIPE-178 definition
! with updated root server networks
! A and J root servers
access-list 180 permit ip host 198.41.0.0 host 255.255.255.0
! B root server
access-list 180 permit ip host 128.9.0.0 host 255.255.0.0
! C root server
access-list 180 permit ip host 192.33.4.0 host 255.255.255.0
! D root server
access-list 180 permit ip host 128.8.0.0 host 255.255.0.0
! E root server
access-list 180 permit ip host 192.203.230.0 host 255.255.255.0
! F root server
access-list 180 permit ip host 192.5.4.0 host 255.255.254.0
! G root server
access-list 180 permit ip host 192.112.36.0 host 255.255.255.0
! H root server
access-list 180 permit ip host 128.63.0.0 host 255.255.0.0
! I root server
access-list 180 permit ip host 192.36.148.0 host 255.255.255.0
! K root server
access-list 180 permit ip host 193.0.14.0 host 255.255.255.0
! L root server
access-list 180 permit ip host 198.32.64.0 host 255.255.255.0
! M root server
access-list 180 permit ip host 202.12.27.0 host 255.255.255.0
access-list 180 deny ip any any
access-list 181 permit ip any 255.255.255.0 0.0.0.255
access-list 181 deny ip any any
access-list 182 permit ip any 255.255.252.0 0.0.3.255
access-list 182 deny ip any any
access-list 183 permit ip host 169.223.0.0 host 255.255.0.0
access-list 183 deny ip any any

```

IP Prefix-List Example

Prefix-lists can also be used – the above example has been rewritten using the new ip prefix-list commands available in 11.1CC and 12.0 software releases. This makes the configuration more readable, if not more intuitive. Again, remember to check the root nameserver networks for any changes before implementing these. It is also worth checking the BGP routing table to ensure that these networks are still announced with the prefix lengths listed below.

```

router bgp 65280
  bgp dampening route-map RIPE178-flap-dampen
  !
  ip prefix-list my-nets description Networks we don't suppress
  ip prefix-list my-nets seq 5 permit 169.223.0.0/16
  !
  ip prefix-list suppress22 description Dampening of /22 and /23 prefixes
  ip prefix-list suppress22 seq 5 permit 0.0.0.0/0 ge 22 le 23

```

Saturday, February 19, 2000

```
!  
ip prefix-list suppress24 description Dampening of /24 and longer prefixes  
ip prefix-list suppress24 seq 5 permit 0.0.0.0/0 ge 24  
!  
ip prefix-list rootns description Root-nameserver networks  
ip prefix-list rootns seq 5 permit 198.41.0.0/24  
ip prefix-list rootns seq 10 permit 128.9.0.0/16  
ip prefix-list rootns seq 15 permit 192.33.4.0/24  
ip prefix-list rootns seq 20 permit 128.8.0.0/16  
ip prefix-list rootns seq 25 permit 192.203.230.0/24  
ip prefix-list rootns seq 30 permit 192.5.4.0/23  
ip prefix-list rootns seq 35 permit 192.112.36.0/24  
ip prefix-list rootns seq 40 permit 128.63.0.0/16  
ip prefix-list rootns seq 45 permit 192.36.148.0/24  
ip prefix-list rootns seq 50 permit 193.0.14.0/24  
ip prefix-list rootns seq 55 permit 198.32.64.0/24  
ip prefix-list rootns seq 60 permit 202.12.27.0/24  
!  
route-map RIPE178-flap-dampen deny 10  
  match ip address prefix-list my-nets  
!  
route-map RIPE178-flap-dampen deny 20  
  match ip address prefix-list rootns  
!  
route-map RIPE178-flap-dampen permit 30  
  match ip address prefix-list suppress24  
  set dampening 30 750 3000 60  
!  
route-map RIPE178-flap-dampen permit 40  
  match ip address prefix-list suppress22  
  set dampening 15 750 3000 45  
!  
route-map RIPE178-flap-dampen permit 50  
  set dampening 10 1500 3000 30  
!
```

APPENDIX 3 – TRAFFIC ENGINEERING TOOLS

Internet Traffic and Network Engineering Tools

As a follow-up on how to track where your customer are going on the Net, here is a list of tools on the Internet that can be used to pull in statistics from your network. Most ISP do not use things like HP OpenView, SunNet Manager, Cisco Works, or Spectrum to manage their networks. These network management packages are great for the enterprise LANs, but do not have the simple scaleable tools needed for ISP networks. Instead, ISPs pull together different, mostly public domain, tools and use UNIX scripts to generate charts and reports (via GNU Plot) for traffic engineering and Quality of Service management.

Note that one Cisco specific solution is to use NetFlow. NetFlow is available on 75XX, 72XX and 7000/RSP running the 11.1CC, 12.0 and later software releases. It is available on the 3600 series and higher platforms from 12.0(1) and is available on the smaller platforms from 12.0(2)T. There are a few white papers on the Cisco Web pages. Check them out and/or let the authors know if you would like more information.

Stan Barber gave a talk at the February NANOG entitled *Monitoring your Network with Freely Available Statistics Reporting Tools*. The slides are at <http://www.academ.com/nanog/feb1998/nettools.html>.

CAIDA

CAIDA (Co-operative Associations for Internet Data Analysis) has a very comprehensive page listing a lot of tools and pointers. This CAIDA effort is supported through assistance provided by US National Science Foundation and Cisco Systems.

CAIDA Measurement Tool Taxonomy <http://www.caida.org/Tools/taxonomy.html>

NetScarf/Sicon

New project by MERIT to get a picture on what is happening on the Internet. They now have an NT version:

<http://nic.merit.edu/~netscarf/>

NeTraMet/NetFlowMet

The old one and one of the best for TCP/IP flow analysis. SingNet used NeTraMet on an Intel PC with BSD Unix and a Digital FDDI card. The results were dumped on to a system that did all the flow analysis and the results were posted on to an internal Web server. Recently a capability has been added to analyse Cisco NetFlow records – NetFlowMet is part of this package now.

<http://www.auckland.ac.nz/net/NeTraMet/>

Cflowd

One of the better tools out there for NetFlow analysis is cflowd. Our customers developed this tool for their own use. It is free and located at:

<http://www.caida.org/Tools/Cflowd/>

Other scripts based on cflowd are located at:

<http://buckaroo.xo.com/CFLOWD/>

Saturday, February 19, 2000

The key Cisco documents on NetFlow are constantly being updated (because we are adding new features and functionality all the time). Do a keyword search on CCO to find all the documentation on NetFlow.

NetFlow tools (flowdata.h, fdrecorder.c, fdplayback.c, fdg.c) that were used to build cflowd are located on Cisco's FTP site:

<ftp://ftp-eng.cisco.com/ftp/NetFlow/fde/README>

MRTG

Multi Router Traffic Grapher – a Perl script based package to create graphics of interface loading on the routers. Saves you having to create UNIX scripts to do the same thing! It works on UNIX (requires the more recent versions of Perl 5) and there is a version for **Windows NT** too! The package also contains contributed tools allowing you to monitor CPU loads, disk space, temperature, and many other functions which an ISP can use to watch its network.

<http://www.mrtg.org>

<http://ee-staff.ethz.ch/~oetiker/webtools/mrtg/mrtg.html>

<http://mailer.gu.se/traffic/mrtg.html>

RRDTool

RRDTool is from the author of MRTG, and is designed to be a more powerful and flexible system for graphing collected statistics. It is not meant to be a full replacement for MRTG, and future versions of MRTG are planned to sit on top of RRDTool.

<http://ee-staff.ethz.ch/~oetiker/webtools/rrdtool/>

Vulture

SNMP Vulture is a tool to do long term SNMP data collection and analysis of routers and other similar devices. Vulture has a number of features that make it suitable for a number of different tasks:

- Per-interface configuration. Different data may be collected on each defined interface.
- Template based configuration. Different sorts of interfaces may require recording different information – whoever heard of collisions on a serial interface?
- Configurable per-router community strings.
- Web based graphical browsing of stored data.
- Built-in data archival mechanism for stale data.

Vulture is written in Perl version 5, and uses the CMU version 2 libraries to do the low level snmp access. The Vulture distribution includes both the CMU libraries and a small module to connect the libraries to Perl. The browser interface also requires the generally available Gnuplot and PBMPlus utilities to generate graphical output.

<http://www.vix.com/vulture/>

CMU SNMP

Free command line SNMP software. Create your own scripts to pull down the MIB variables you wish to look at (like information on the links).

<http://www.boutell.com/lsm/lsmbyid.cgi/000033>

http://hpx.petech.ac.za/hppd/hpux/Networking/Admin/cmu_snmp-1.2u/

CMU SNMP Archives – <ftp://lancaster.andrew.cmu.edu/pub/snmp-dist/>

UCD SNMP (the successor to CMU SNMP)

CMU SNMP has not been updated for some time now. UCD has taken over the project and their release contains a port and modified code of the CMU 2.1.2.1 snmp agent. It has been modified to allow extensibility quickly and easily. It is far from the best and most configurable system; but hey, it is free.

<ftp://ftp.ece.ucdavis.edu/pub/snmp/ucd-snmp.README>

<ftp://ftp.ece.ucdavis.edu/pub/snmp/ucd-snmp.tar.gz>

Gnuplot

Gnuplot is a useful public domain graphing tool. If configuring MRTG is too much, or you need to graph something else quickly, this is probably the way to do it.

http://www.cs.dartmouth.edu/gnuplot_info.html

An example of what can be done is on the NOAA web site:

<http://www.erl.noaa.gov/network.html>

<http://www.erl.noaa.gov/ahsia/webshop/overview.html>

A Comprehensive list of other Public Domain SNMP Software can be found at:

<http://wwwsnmp.cs.utwente.nl/ietf/impl.html>

NETSYS

NetSys is a suite of tools developed to map, track, and **predict** problems on your network. People who use it really like its capabilities of mapping your existing network and then working projections of what will happen to the links under specific conditions. Here are more details on the NETSYS suite of tools:

NETSYS Connectivity Tools. The first in a series of simulation-based planning and problem-solving products for network managers, analysts, and designers. The Connectivity Tools assist network planners with problem solving, design, and planning activities focusing on network connectivity, route, and flow analysis.

http://www.cisco.com/warp/public/734/nslms4/cnmgr_ds.htm

NETSYS Performance Tools. These tools allow users to create a network baseline from configuration and performance data, and then analyze the interactions between traffic flow, topology, routing parameters, and Cisco IOS features. Users can also diagnose and solve operational problems, test "what if" scenarios, tune the network configurations for improved performance, and plan for incremental network changes.

http://www.cisco.com/warp/public/734/nslms4/pfmgr_ds.htm

NETSYS Advisor. NETSYS Advisor works with both the Connectivity and Performance Tools to quickly isolate network problems and identify solutions using an exclusive model-based reasoning technology.

<http://www.cisco.com/warp/public/734/nslms4/>

Saturday, February 19, 2000

SysMon

Sysmon is a network monitoring tool designed by Jared Mauch to provide high performance and accurate network monitoring. Currently supported protocols include SMTP, IMAP, HTTP, TCP, UDP, NNTP, and PING tests. The latest version can be found at:

<http://puck.nether.net/sysmon/>

Treno

Treno (a tools develop end to end performance) information and you can try it out from PSC via this WWW forms interface:

<http://www.psc.edu/~pscnoc/treno.html>

Scotty – Tcl Extensions for Network Management Applications

Scotty is the name of a software package which allows to implement site specific network management software using high-level, string-based APIs. The software is based on the [Tool Command Language](#), which simplifies the development of portable network management scripts. The scotty source distribution includes two major components. The first one is the Tnm Tcl Extension, which provides access to network management information sources. The second component is the Tkined network editor which provides a framework for an extensible network management system.

This tool used to be called “tkined”. Scotty is now the ‘correct’ name for the project.

<http://wwwsnmp.cs.utwente.nl/~schoenw/scotty/>

With all of this software available, it is *not* expensive or difficult to collect and analyse data on your network. You can create your own tools and run them on the many Intel based UNIX systems (i.e. Linux, BSDI, etc.).

Other Useful Tools to Manage your Network

RTRMon – A Tool for Router Monitoring and Manipulation

The RTR system currently comes with three programs, rtrmon, rtrpass, and rtrlogin. rtrmon – for “router monitor” – is the core of the system, using predefined actions to log in to routers, issues a command, process the output, archive the result, and possibly mail reports. It is designed to provide the framework for a variety of potential monitoring tasks and be readily extensible with new reporting code if the built-in methods are insufficient for complex analysis. It can even update router configurations, despite its “monitor” moniker.

The rtrpass program is meant to provide an easy interface to a more secure method of storing passwords. Since rtrmon needs to be able to provide passwords to routers to log in to them and to gain “enable” privileges, they have to be accessible on the hosts on which rtrmon is running. To reduce the risks associated with this, rtrpass manages a PGP-encrypted file for each rtrmon user that contains his own password and his enable password, if he has enable access. Passwords can be controlled on a per router basis if desired.

rtrlogin is used to login to routers, automatically logging in using your username and password, getting “enable” privileges, setting the terminal length and width to the size of your window, and running any commands you have saved in your personal login file. The session can then be used interactively.

<http://www.vix.com/rtrmon/>

Cisco’s MIBs

All the Cisco SNMP MIBs are publicly available. If you have commercial SNMP management packets and/or shareware-freeware packets, you may need to go and grab the MIB. Here is the FTP site:

<ftp://ftp.cisco.com/pub/mibs/>

SECURE SYSLOG (ssyslog)

SECURE SYSLOG (ssyslog) is available for UNIX systems. Designed to replace the syslog daemon, ssyslog implements a cryptographic protocol called PEO-1 that allows the remote auditing of system logs. Auditing remains possible even if an intruder gains superuser privileges in the system, the protocol guarantees that the information logged before and during the intrusion process cannot be modified without the auditor (on a remote, trusted host) noticing.

<http://www.core-sdi.com/ssyslog>

Overall Internet Status and Performance Tools

Many people are asking just how well the Internet is performing. Ever since the NSFnet was decommissioned, there has been no one place to understand the performance and traffic profiles on the Internet. Yet, there are people trying to figure out how to do this. The following lists are sites, projects, and software that is attempting to get a true “big picture” of what is happening on the Internet. ISPs can elect to join one or more of these programs to add more data to these projects.

NetStat

A tool that pings various parts of the Internet from various locations on the Internet, collects the data, and provides an average response time on the major US backbone. The tool is based on ping.

<http://netstat.net/>

What other ISPs are doing...

Here are examples of what ISPs from all over Internet are using to manage their network. Randy Bush randy@psg.com asked major ISPs in the US on the NANOG mailing list what they used for traffic analysis. Here is Randy’s summary. Notice the number of UNIX script based tools...

We do SNMP polling ever 15 minutes at SESQUINET on every line over which we have administrative control and over every peering point. We produce a daily report on errors and usage. We are getting ready to switch to Vulture or NetScarf (or some combo) to give us more interactive information.

We perform measurement of certain basic network parameters, such as usage (bandwidth used / total bandwidth) and line error rates on all of our non-customer links. We perform CPU usage, memory usage, and environmental monitoring of all our routers. We also perform the line usage and error rate on all customer lines. We monitor all of our customers' routers unless they say otherwise, and notify them of any problems.

Saturday, February 19, 2000

Finally, we monitor select points throughout the Internet (root name servers, etc.) on a 4 times an hour basis using pings. We accomplish this monitoring using the following items: an in-house built package that uses SNMP, traceroute, and ping to provide graphs and tabular statistical information. We use Cabletron's Spectrum for a quick network overview.

We do SNMP MIB-II stuff, plus the cflowd stuff and something we call 'mxd' (measures round trip times, packet loss and potential reason, etc. from a whole bunch of different points in our network to a bunch of other points in our network... We use it to create delay matrices, packet loss reports and other reports). There are some other things, but these are the biggies.

The mxd thing was originally just sort of a toy for neat reports, but in the last year it's become a critical tool for measuring delay variance for one of our VPDN customers that does real-time video stuff (and is to some extent helping us figure out where we've got delay jitter and why; on the other hand it's also raising more questions ☺).

Since most of my professional career has been in the enterprise world, I can offer you what we used to measure availability to our mail servers, web servers, DNS servers, etc., at one of my previous employers.

We employed several application tests, along with network performance tests. Our primary link was via UUnet, a burstable T1. We purchased an ISDN account from another local provider, who wasn't directly connected to UUnet. Probably a good example of a joe-average-user out there.

Every 5 minutes, we measured round-trip response times to each of the servers and gateway router (via ping) and recorded it. We also had application tests, such as DNS lookups on our servers, timing sendmail test mails to a /dev/null account, and time to retrieve the whole home page.

This wasn't meant to be a really great performance monitoring system; it was actually meant to 1) check how our availability looked from a "joe user" perspective on the net (granted, reachability/availability wasn't perfect because it was only one point in the net) and 2) look at response time trends / application trends to see if our hardware/software was cutting it.

We use a traffic flow monitoring system from Kaspia Systems. (www.kaspia.com) The Kaspia product collects all sorts of data from router ports and RMON probes, stores the data and performs various trend analysis. We collect traffic flow, router CPU usage and router memory information plus various errors. There is a data reduction process which runs once a day, and a very nifty web interface. The product is not cheap, but the system definitely fills a void here.

Maybe I should organize a talk on what we are doing with it for an upcoming NANOG? As an old instrumentation engineer, I think the basis of our use of the tool is pretty solid. Plus, I actually developed a means for calibration of the accuracy of the flow data. Have not had time yet to work out a validation for the trends, but I'll get to it one of these decades.

Also, the Kaspia people will give you a thirty day trial on their product at no charge.

For non-intrusive stuff, we keep a log of all interface status changes on our routers, and we pull five-minute byte-counts inbound and outbound on each interface, which we graph against port speed. Watching the graphs for any sort of clipping of peaks gives a pretty good indication of problems, and watching for shifts of traffic between ports on parallel paths likewise.

As for intrusive testing, we do a three-packet min-length ping to the LAN-side port of each of our customers' routers once each five minutes, and follow that up with additional attempts if those three are lost. We log latency, and if we have to follow up with a burst, we log loss rate from the burst. Pinging through to the LAN port obviously lets us know when CPE routers konk out, as occasionally we see hung routers that still have

operational WAN ports talking to us, likewise, simply watching VC-state isn't a reliable enough indicator of the status of the remote router. Plus it tells you if the customer has kicked the Ethernet transceiver off their equipment, for instance. Wouldn't matter to you, probably, but our demarc is all the way out at their WAN port, since we own and operate our customers' CPE.

I think a bit about what more we could be doing; flows-analysis and whatnot... It's nice to think about, and eventually we'll get around to it, but programmer-time is relatively precious, and other things have higher priority, since the current system works and tends to tell us most of what we seem to need to know to provide decent service.

We place quite a bit of emphasis on network stats. currently we have about 3 years of stats online, and are working on converting our inhouse engine to an rdbms so we can more easily perform trend analysis. besides kaspia, other commercial packages include trendsnmpp (www.desktalk.com) and concord's packages (www.concord.com). Our in house stuff is located at <http://netop.cc.buffalo.edu/> if you are curious about what we do.

We are Neanderthals right now – we use a hacked rcisco to feed data to nocol. We watch bandwidth (separately as well) on key links - and also watch input errors and interface transitions (for nocol) – all done with perl and expect-like routines, parsing 'sho int's every few minutes.

Emergency stuff goes through nocol; bandwidth summaries are mailed to interested parties overnight.

We have running here now the MRTG package that generate some fancy graphics, but in my opinion these graphics are useless and looking in detail to some of the reports they are not accurate, several of our clients request the raw data but this package only maintain few raw data just to generate the graphs, mean also useless.

In the past we use to have also a kind of ASCII reports (Vikas wrote some of the scripts and programs) generated from information obtained using the old snmp tool set developed by nysernet but I guess that nobody maintained the config files and I believe that the snmp library routines used aren't working fine.

So, I need to invest some time to provide a fast solution to this, I'll appreciate your help to identify some useful package or directions about how to generate some good looking and consistent reports.

We have been using the MRTG package which is basically a special SNMP agent that queries the routers for stats and then does some nice graphing of the data on the web.

SNMP queries with a heavily-modified version of MRTG from the nice guy in Germany. Works very nicely. We have recently installed NetScarf 2.0, and are contemplating merging NetScarf 3.0 with the MRTG front end.

I'm researching whether I can rewrite Steve Corbato's fastpoll program using the fastsnmp library from the NetScarf people. I think this will allow fastpoll to scale better. I've successfully written a quick C program that uses the library to collect the required data for a router – now I've just got to make it so we can manage it easily (i.e. auto-generated config files from our databases).

My goal is to be able to collect 1-2 minute period data on all links that are greater than 10 Mbps – 15 minutes data for everything else. The 2 minute collection period will allow to scale up to 280Mbps before experiencing two counter roll-overs within a polling interval. Hopefully that will hold us over the interface counters are available as Counter64 objects via SNMPv2 (if that ever happens).

BTW – what fastpoll collects now is ifInOctets, ifOutOctets, ifInUcastPkts, ifOutUcastPkts, ifInErrors and ifOutDiscards. Rather than storing the raw counters, it calculates the rate by taking the delta and dividing by the period. Getting the accurate period is actually the hard part – I am having SNMP send me the uptime of the

Saturday, February 19, 2000

router in each query and using that to calculate the interval between polls and to detect counter resets due to reboots. The other trick to handle is the fact that, while IOS updates the SNMP counters for process-switched packets as they are routed, it looks like the counter for SSE switched packets on C70X0 routers only get updated once every 10 seconds.

APPENDIX 4 – EXAMPLE ISP ACCESS SECURITY MIGRATION PLAN

What follows is one example of how an ISP can migrate their network equipment (routers, switches, and NAS) from a state where telnet access is open to the outside world to the point where only specific authorised workstations are allowed access to the telnet prompt.

Unfortunately, at the time this version was written, most ISPs still do not take these simple precautions to help secure their network. This section is specifically addressed to those ISPs who have not taken the extra time to undertake the basics. A simple procedure that draws a security circle around an ISP's network and then slowly narrows the circle tighter and tighter until just the authorised IP addresses are included in the VTY's ACL.

Phase One – Close off access to everyone outside your CIDR block.

The first step for any ISPs securing their network is to close telnet access off to everyone but authorised workstations. This is done in phases. The first phase is to create a standard ACL that would permit telnet from IP addresses in the ISP's CIDR block. This ACL is used with the VTY's `access-class` command to ensure the source IP address of any telnet packet coming to the VTY port matches the ACL.

Why just the ISP's CIDR block? First, it's easy for an ISP who is not doing this technique now, to implement. The ISP does not have to worry about locking out staff members who need access to the router from different parts of the network. Hence, it can be done with the least worry that it will effect the ISP's operations. Second, it limits the threat for the entire Internet to just IP addresses inside the ISP's CIDR block. Minimising risk is one of the fundamental tenants of security. Finally, since the ISP is beginning with no protection, this incremental step insures everything is working before deepening the security configurations on the network. Phase One's theme is to get the ball rolling. Limiting access to just those IP addresses in the ISP's CIDR block gets the ball rolling.

Figure 16 is an example of an ISP's network. The allocated CIDR blocks are 169.223.0.0/16 and 211.255.0.0/19²⁰. Phase One is to create an ACL that can be used with the VTY's `access-class` command to restrict telnet to IP addresses within the CIDR block.

```

aaa new-model
aaa authentication login ISP local
!
username Cisco1 password 7 11041811051B13
!
access-list 3 permit 211.255.0.0 0.0.31.255
access-list 3 permit 169.223.0.0 0.0.255.255
access-list 3 deny any
!
line vty 0 4
 access-class 3 in
 exec-timeout 5 0
 transport preferred none
 login authentication ISP
 history size 256

```

Example 1 – Simple VTY Config to limit access to just the ISP's CIDR Block

The configuration in Example 1 is used on all routers in the network. Similar configurations should be used on the switches in the network. Any staff workstations/servers should also use appropriate tools to limit telnet access to the workstations/server's resources²¹.

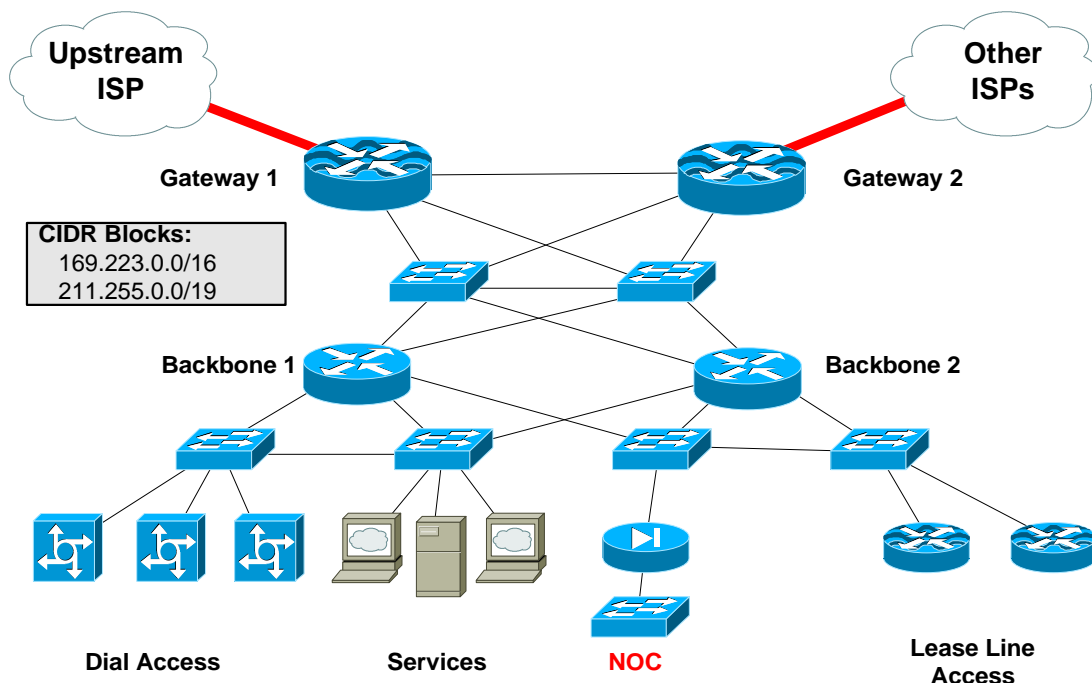


Figure 16 – ISP Network Example

Phase Two – Add Anti-Spoofing Filters to your upstream gateways and peering points.

Limiting the security risk through restricted telnet access is only the first step. The next step would be to insure parties outside the ISP's network would not be able to *spoof* source addresses from the ISP's CIDR block. There are several forms of source address spoofing and telnet sequence number hijacking attacks that can penetrate the VTY's access-class protections. To minimise the risk of these sort of attacks, an ISP can place anti-spoofing filters at the edges of their network.

As highlighted in the section on Egress and Ingress Filtering, anti-spoofing filters are used to insure that any address coming from the Internet into the ISP's network does not contain a source address from the ISP's network. For example, if a packet from the Internet with a source address of 211.255.1.1 comes into the ISP in Figure 17, the anti-spoofing filter would drop the packet.

Remember! No one from the general Internet should be sending you packets with a source address from your own network!

Few ISPs implement anti-spoofing packet filters. The key reason given is possible performance impact. Yes, applying packet filters to any router may cause a performance impact. Yet, another essential tenant of security is balancing the trade offs. Sacrificing some performance to minimise the security risk to valuable network resources²² is a logical trade off. Especially with the new improvements in packet per second (PPS) that the latest Cisco IOS code offers ISPs.²³

²⁰ Apologies if these network represent a real allocation. They were pulled from the APNIC blocks as an example, not to portray any real life network.

²¹ Tools like TCP Wrapper are well known and have a role in an ISP's overall security architecture.

²² Spoofing attacks are more likely to target workstation and server resources. These resources would likely depend on tools like TCP Wrapper – these wrappers can also be bypassed by spoofing attacks. So, placement of anti-spoofing filters protects the entire network, not just the routers.

²³ Distributed Switching (FIB), Cisco Express Forwarding (CEF), Distributed NetFlow, and other improvements in the 11.1CC and 12.0S code trains.

Where to place the anti-spoofing packet filters?

Place anti-spoofing filters at the edge of an ISP's network. This usually means the router(s) that interconnect with other ISPs. Routers *Gateway1* and *Gateway2* are examples in Figure 17. Any attempt at spoofing from the core of the Internet (i.e. the upstream ISP) and/or an ISP peer connection would be dropped on the inbound interface.

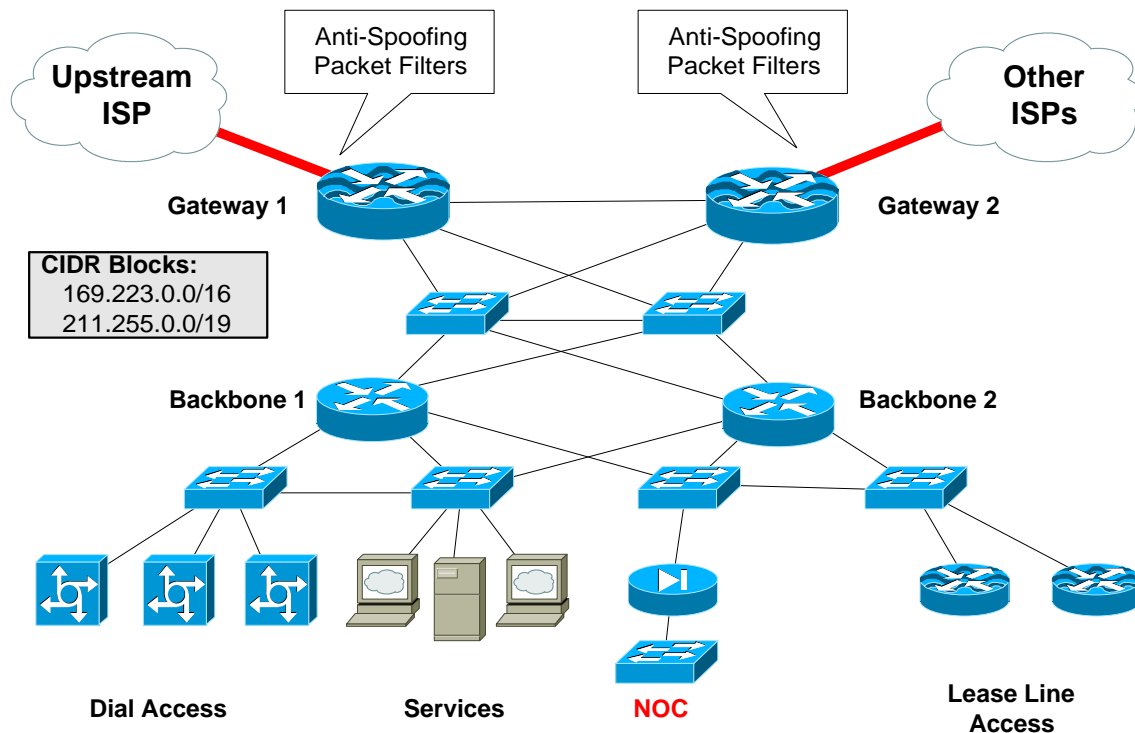


Figure 17 – Applying Anti-Spoofing Filters

Example 2 highlights a typical anti-spoofing filter. Notice that there are two CIDR blocks used in the example – 169.223.0.0/16 and 211.255.0.0/19 two lines of seven in the ACL 111. Here is what the other lines do:

- `deny ip 127.0.0.0 0.255.255.255` – This is the loopback address for TCP workstations, PCs, and servers. It should not be transmitted over the Internet. If it is, then it is either a broken TCP stack or someone trying to break into a resource.
- `deny ip 10.0.0.0 0.255.255.255` – RFC 1918 Private Address Space. Cisco advocates RFC 1918 Private Address Space use in enterprise networks in conjunction with Network Address Translation (NAT). Any packets with RFC1918 addresses in their source are either from a broken NAT implementation or part of a spoofing attack.
- `deny ip 172.16.0.0 0.15.255.255` – RFC 1918 Private Address Space. Ibid.
- `deny ip 192.168.0.0 0.0.255.255` – RFC 1918 Private Address Space. Ibid.
- `deny ip 169.223.0.0 0.0.255.255` – One of the example ISP's CIDR blocks.
- `deny ip 211.255.0.0 0.0.31.255` – The other CIDR Block.
- `permit ip any any` – Permit normal packets.

Saturday, February 19, 2000

Each line with a *deny* has a *log* option turned on. This will take any matches and include them in the internal and output to syslog (if the router has it configured). The *log* option is not used on the last *permit*. Good packets do not need to be logged.

```
Router Gateway1
!
interface hssi 0/1
  description 16Mbps link to our upstream provider
  bandwidth 16384
  ip access-group 111 in
  no ip redirects
  no ip directed-broadcast
  no ip proxy-arp
!
access-list 111 deny ip 127.0.0.0 0.255.255.255 any log
access-list 111 deny ip 10.0.0.0 0.255.255.255 any log
access-list 111 deny ip 172.16.0.0 0.15.255.255 any log
access-list 111 deny ip 192.168.0.0 0.0.255.255 any log
access-list 111 deny ip 169.223.0.0 0.0.255.255 any log
access-list 111 deny ip 211.255.0.0 0.0.31.255 any log
access-list 111 permit ip any any

Router Gateway2
!
interface serial 0/1
  description Compressed 2Mbps bi-lateral peer to our neighboring country
  bandwidth 2048
  ip access-group 111 in
  no ip redirects
  no ip directed-broadcast
  no ip proxy-arp
!
access-list 111 deny ip 127.0.0.0 0.255.255.255 any log
access-list 111 deny ip 10.0.0.0 0.255.255.255 any log
access-list 111 deny ip 172.16.0.0 0.15.255.255 any log
access-list 111 deny ip 192.168.0.0 0.0.255.255 any log
access-list 111 deny ip 169.223.0.0 0.0.255.255 any log
access-list 111 deny ip 211.255.0.0 0.0.31.255 any log
access-list 111 permit ip any any
```

Example 2 – Anti-Spoofing Configuration Example

Phase Three – Close off network equipment access to everyone except the NOC and other authorised staff

Once Phase One and Phase Two is completed, work can begin on Phase Three – closing off access to everyone except the NOC Staff. The speed from which an ISP moves from the first two phases to the third phase is dependent on the confidence of the ISP's engineers. Some may wish to monitor the operational impact of the first two phases before incrementing another layer of security. Others may bypass Phase One and immediately restrict access only to the NOC Staff. Either way works. Each ISP needs to develop plans that suite their environment.

There are two basic steps to necessary to close telnet access to just the NOC Staff. First, identify the IP address block for the NOC's network. Figure 18 shows the NOC's network behind a firewall using IP block 211.255.1.0/24. Second, modify the ACLs for the VTY's access-class with the addresses assigned to the NOC. Example 3 highlights this modification.

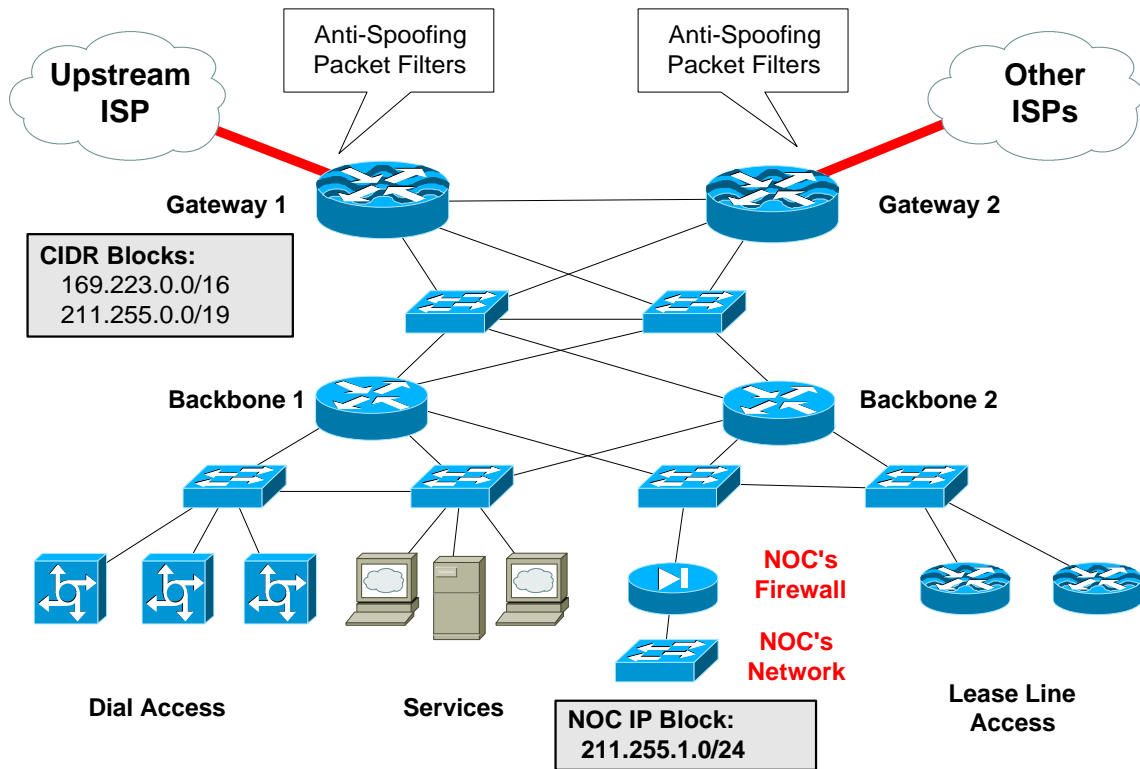


Figure 18 – Closing off access to everyone except the NOC Staff

```

aaa new-model
aaa authentication login ISP local
!
username Cisco1 password 7 11041811051B13
!
access-list 3 permit 211.255.1.0 0.0.0.255
access-list 3 deny any
!
line vty 0 4
access-class 3 in
exec-timeout 5 0
transport preferred none
login authentication ISP
history size 256
    
```

Example 3 – ACLs with Telnet Access closed to all but the NOC's Network