JUNIPER NETWORKS

AMBASSADOR

Junos® Networking Technologies

DAY ONE: JUNIPER AMBASSADORS' COOKBOOK FOR ENTERPRISE

The Juniper Ambassadors take on the top support issues and then some, in this book of over a dozen solutions for the Junos Enterprise user.

By Martin Brown, Ben Dale, Glen Kemp, Petr Klemaj, Chris Jones, Steve Puluka, Michel Tepper, and Scott Ware

DAY ONE: JUNIPER AMBASSADORS' COOKBOOK FOR ENTERPRISE

The Juniper Ambassador program recognizes and supports its top community members and the generous contributions they make through sharing their knowledge, passion, and expertise on J-Net, Facebook, Twitter, and other social networks. The Juniper Ambassadors are a diverse set of network engineers, consultants, and architects who work in the field with Juniper technologies on a daily basis.

In this *Day One* cookbook for Enterprise network administrators, the Juniper Ambassadors take on some of the top support issues and provide clear-cut solutions and frank discussions on how to keep things running. From creating an aggregate link between a Juniper and Cisco switch, to deploying IPsec VPN using Junos Space, to connecting two networks using a SRX Series, the sixteen recipes in this cookbook are meant to provide quick and tested solutions to everyday issues.

IT'S DAY ONE AND YOU HAVE A JOB TO DO, SO LEARN HOW TO:

- Create a Junos image that can be deployed from a USB thumb drive.
- Set up highly-available VPN connections that fail-over quickly with bi-directional forwarding.
- Use routing instances to disable split tunnelling and prevent branch offices from directly to the Internet.
- Enable security and multicast on your OSPF instances to prevent interference by unauthorized devices.
- Use the Host Checker to differentiate between the various security postures of devices connecting to your network.
- Use IP tracking and Real-time Performance Monitoring to determine whether the primary or backup ISP link should be used.
- Configure the Multicast Bootstrap router protocol using IPv4 and IPv6.
- Use VPN tunnelling policies to control connectivity for Network Connect and Junos Pulse clients.
- ... and much more.

Published by Juniper Networks Books

ISBN 978-1936779697

52000

9 781936 779697

07100171

Day One: Juniper Ambassadors' Cookbook for Enterprise

By Martin Brown, Ben Dale, Glen Kemp, Petr Klemaj, Chris Jones, Steve Puluka, Michel Tepper, and Scott Ware

JUNIPer NETWORKS

Published by Juniper Networks Books
Authors: Martin Brown, Ben Dale, Glen Kemp, Petr Klemaj, Chris Jones, Steve Puluka, Michel Tepper, and Scott Ware
Technical Reviewer: Kevin Barker
Editor in Chief: Patrick Ames
Copyeditor and Proofer: Nancy Koerbel
J-Net Community Manager: Julie Wider

ISBN: 978-1-936779-69-7 (print)
Printed in the USA by Vervante Corporation.

ISBN: 978-1-936779-70-3 (ebook)

Version History: v1, July 2013
2 3 4 5 6 7 8 9 10 #7100164-en

This book is available in a variety of formats at: http://www.juniper.net/dayone. Send your suggestions, comments, and critiques by email to dayone@juniper.net.

Welcome to Day One

This book is part of a growing library of *Day One* books, produced and published by Juniper Networks Books.

Day One books were conceived to help you get just the information that you need on day one. The series covers Junos OS and Juniper Networks networking essentials with straightforward explanations, step-by-step instructions, and practical examples that are easy to follow.

The *Day One* library also includes a slightly larger and longer suite of *This Week* books, whose concepts and test bed examples are more similar to a weeklong seminar.

You can obtain either series, in multiple formats:

- Download a free PDF edition at http://www.juniper.net/dayone.
- Get the ebook edition for iPhones and iPads from the iTunes Store. Search for Juniper Networks Books.
- Get the ebook edition for any device that runs the Kindle app (Android, Kindle, iPad, PC, or Mac) by opening your device's Kindle app and going to the Kindle Store. Search for Juniper Networks Books.
- Purchase the paper edition at either Vervante Corporation (www.vervante.com) or Amazon (amazon.com) for between $12-$28, depending on page length.
- Note that Nook, iPad, and various Android apps can also view PDF files.
- If your device or ebook app uses .epub files, but isn't an Apple product, open iTunes and download the .epub file from the iTunes Store. You can now drag and drop the file out of iTunes onto your desktop and sync with your .epub device.

Welcome Ambassadors!

Juniper Ambassadors are global technical/brand advocates that actively participate across Juniper community and social programs. They are a diverse set of network engineers, consultants, and architects who work in the field with Juniper technologies on a daily basis. The Juniper Ambassadors' mission is spreading the word about the power of Junos to the world's networking and security engineers. Welcome Ambassdors!

Audience

This book is intended for Enterprise network administrators and provides field-tested recipies for common network deployment scenarios, as well as brief background information needed to understand and deploy these solutions in your own environment.

This book's recipes are numbered, and they are loosely organized into the following topic areas:

- Switching: 1 and 13
- IPsec VPN: 2, 4, 6, 14, 15
- Routing: 5. 9
- Dual ISP: 8, 12
- Junos Operations: 3, 11
- IVE/SSL VPN: 7, 10, 16

What You Need to Know Before Reading

Before reading this book, you should be familiar with the basic administrative functions of the Junos operating system, including the ability to work with operational commands and to read, understand, and change Junos configurations. There are several books in the *Day One* library on exploring and learning Junos available at www.juniper.net/dayone. This book also includes recipes for the IVE operating system on Juniper SSL VPN and Junos Space that require only basic knowledge of network administrative tasks.

This book makes a few assumptions about you, the reader, and presumes you have a:

- Broad understanding of TCP/IP
- Basic knowledge of Ethernet switching concepts, such as bridging and Spanning Tree
- Solid understanding of basic IP, firewalls, routing, and switching concepts
- Familiarity with configuring a variety of network equipment
- Familiarity with common networking protocols and sessions such as IPsec, SSH, Telnet, and HTTPs
- Understanding of basic network administration tasks

After Reading This Book, You'll Be Able To:

- Create an aggregated link between a Juniper and a Cisco switch.
- Deploy IPsec VPNs using Junos Space.
- Create a Junos image that can be deployed from a USB thumb drive.
- Set up highly-available VPN connections that fail-over quickly with bi-directional forwarding.
- Use routing instances to disable split tunnelling and prevent branch offices from directly connecting to the Internet.
- Enable security and multicast on your OSPF instances to prevent interference by unauthorized devices.
- Use the Host Checker to differentiate between the various security postures of devices connecting to your network.
- Use IP tracking and Real-time Performance Monitoring to determine whether the primary or backup ISP link should be used.
- Configure the Multicast Bootstrap router protocol using IPv4 and IPv6.
- Use VPN tunnelling policies to control connectivity for Network Connect and Junos Pulse clients.
- Create a simple SLAX script to create custom commands within Junos.
- Explore the options of available for providing Destination NAT for sites with multiple ISP connections.
- Master template configurations to easily make changes en-masse to Junos.
- Connect two networks using Juniper SRX and a Cisco ASA with a policy-based VPN.
- Overlay OSPF in a hub and spoke VPN.

About the Contributors

Martin Brown is a Network Engineer and Juniper Ambassador with knowledge that covers a broad range of network devices. Martin Brown started his career in IT 20 years ago supporting Macintosh computers and has since progressed to networking. He currently holds JNCIA and JNCIS-ENT and is working on JNCIP-ENT, time permitting.

- Martin's Acknowledgments: There are a few people I would love to thank: my partner Irene, as without her support I wouldn't be in this field of work, my good friend Thomas Soderlund, who inspired me on that flight to Gdansk that I can be so much more, and also to my former boss, David Hunnam, who taught me to believe in myself. I'd also like to extend my thanks to the entire Ambassador team for just being there and sharing the passion we get from doing what we do.

Ben Dale is a Senior Systems Engineer with Comlinx, a Juniper Elite Partner based in Brisbane, Australia. He currently certified JNCIE-SEC #63, JNCIP-ENT and JNCIS-SP and has been working in the networking and systems integration space for more than 13 years. Ben is an active member of J-Net community under the nickname dfex, and on Twitter @labelswitcher.

- Ben's Acknowledgements: I would like to thank my wife and soulmate Donna along with my beautiful daughters Charlotte and Milla for giving me the occasional free time to pursue this incredible hobby/career. I want to also give a big shout out to all the other network engineers out there who give up their time to contribute knowledge and wisdom back to the community via their blogs, forum postings and tweets - we live in an incredible time - keep up the awesome! And finally, a big thanks to Julie Wider, Patrick Ames and the Juniper Champion crew for putting together this team and this book.

Glen Kemp is an Enterprise Security Architect and Juniper Ambassador. He has accumulated numerous Juniper JNCIS and JNCIA qualifications whilst working in Network security for the past 15 years. Glen is also a prolific blogger and writer and works with many leading technologies.

- Glen's Acknowledgements: I must thank my wife Jo and son Samuel for their love and support. A massive thank you must go to Zoe Sands for all her encouragement and wisdom, without which this project could not have happened. Thanks must also go to Jo James for putting me in the right place at the right time and her patience and encouragement. Many others have contributed with knowledge and advice over the years: Geoff Bradley, Patrick Ames, Julie Wider, Graham Duthie, Kev Peterson, Prem Ananthakrishnan, the Packet Pushers illuminati (Greg and Ethan), Rivka Little, and Chuck Moozakis; all of whom deserve massive thanks.

Petr Klemaj is a Juniper Ambassador and a Juniper Networks certified instructor working at Poplar Systems, a Juniper-Authorized Education Partner in Russia. He is certified JNCIE-SEC, JNCIE-ENT, and JNCIP-SP and has several years of experience supporting Juniper equipment for many small and large Juniper customers. He teaches a variety of Juniper classes on a regular basis, beginning with introductory classes such as IJOS and including advanced classes such as AJSEC and JAUT.

- Petr's Acknowledgements: I would like to thank my family and my colleagues at Poplar Systems for their constant support and inspiration.

Chris Jones is a Lead Network Engineer with TorreyPoint and Proteus Networks, certified with Juniper as JNCIE-ENT #272, and with Cisco Systems as CCIE #25655 (R&S). He has a decade of industry experience with both Cisco and Juniper products and solutions, designing and building networks for both small and large enterprises as well as major service providers. Chris is the author of the *Proteus Networks JNCIE-ENT Preparation Workbook*, as well as *Day One: Junos for IOS Engineers*. He was also a technical reviewer for *Junos Enterprise Routing 2nd Edition,* as well as a number of other Juniper *Day One* books. Chris can be found on Twitter @junoschris, and his technical blog is located at http://www.3fives.com. His professional website is http://www.iamchris.ca.

- Chris's Acknowledgments: I would like to thank Patrick Ames and Julie Wider for the opportunity to contribute to this book.

Steve Puluka is a Senior Network Security Engineer with UPMC in Pittsburgh, PA. He is part of a team that manages about 400 firewalls – primarily ScreenOS and Junos, with a Palo Alto presence, and two Cisco VPN router clusters. He holds a BSEET along with the professional level certification in Junos Security and specialist level in ScreenOS and SSL VPN and his original associates in ER and EX. He also has certification and extensive experience in Microsoft Windows server, along with strong VMware skills starting with with Version 2. He has enjoyed supporting networks for more 20 years.

Michel Tepper is a Juniper consultant and instructor working for Westcon Security in the Netherlands. He started working in ICT in 1987. Michel is also is a Juniper Ambassador. Currently he holds three Junos Professional certifications and a number of specialist and associate certifications on non-Junos tracks. Michel is an active member of J-Net and juniperforum.com, where he uses the nickname *screenie* referring to the ScreenOS with which he started his Juniper Journey.

- Michel's Acknowledgements: My thanks and appreciations go out to my students. It's they who teach me what's needed and it's from their questions that I keep learning. Special thanks go out to Valentijn Flik: his technical drawings attain a level I'll never be able to reach. Last but not least: Thank you Julie Wider, for inviting me into the Ambassadors program and starting this book. Thank you Patrick Ames for the many corrections: I learned a lot from you!

Scott Ware is a Network Security Engineer and Juniper Ambassador. He currently holds a JNCIA-Junos certification, and is working on the JNCIS-SEC. Scott has been doing networking and security for over 10 years. You can usually find him hanging around the J-Net forums and on Twitter, if not at an ice rink!

- Scott's Acknowledgements: I would like to thank my lovely Wife Mehgan for all her constant love, encouragement, and support. With her by my side, everything is possible! A very special thank you to Juniper Networks, Julie Wider, and Patrick Ames for helping make this opportunity possible for me. I cannot thank you enough for everything! Many thanks to Chris Jones, Daniel McNulty, Seema Kathuria, Jason Frazier, Tyler Dykema, Great Lakes Computer, and all my fellow Ambassadors for your help and support.

Technical Reviewer, **Kevin Barker,** has over 30 years of IT experience spanning Operations, Applications, and Infrastructure, Kevin is a Co-founder and Chief Technology Officer of Independent Technology Group, a southern California based Juniper Elite partner. Kevin is a frequent contributor to the Juniper user community forums and has achieved the title of Distinguished Expert, J-Net Community and is a member of the Juniper Ambassador program wherein he evangelizes on behalf of Juniper and Junos in the networking and security communities. He is also a JNCIP-SEC, JNICS-ENT, and a Juniper Certified Instructor.

Recipe 1

Aggregated Links

Two of the most important topics in today's network infrastructure are redundancy and how fast the network performs. Aggregated links can offer redundancy and increased throughput between switches at a relatively low cost and without requiring any special equipment.

Having redundant links between switches is a great idea. Should one cable or port fail, the backup link can take over and forward traffic on its behalf. Due to Spanning Tree, the downside is that only one link would be active at any one time, wasting perfectly good ports. To correct this, *aggregated links* were developed, allowing a switch to see several physical ports as a single logical port, fooling Spanning Tree into believing only one connection is present so Spanning Tree doesn't block any ports. Traffic is then load-shared between the links. In addition, should a port or cable fail, the switches will automatically compensate, removing that physical link from the logical link.

Problem

The problem facing network engineers is that different vendors use different methods. For example, AVAYA uses multi-link trunking among other methods, and Cisco uses PAGP. Thankfully, in 2000, the IEEE released 802.3ad, which uses a standard protocol called LACP. Most manufacturers support this in addition to their own standards, and 802.3ad greatly assists engineers when creating aggregated links in multivendor environments.

This recipe illustrates a multi-vendor environment where a Juniper EX Series switch is connected to a Catalyst switch. Ports ge-0/0/12, ge-0/0/13, and ge-0/0/14 on the EX will connect to fa0/1, fa0/2, and fa0/3 on the Catalyst switch, respectively. The link will then be configured to pass traffic for VLAN600 only.

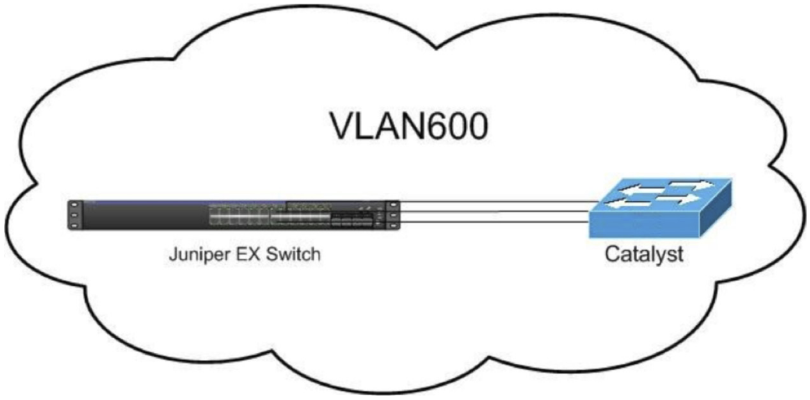


Figure 1.1 This Recipe's Topology

The configuration for both of these devices is set as the default, aside from the host name.

Solution

The first step is to connect the switches at Layer 1 and straight away you'll hit the first problem. The EX has the ports defaulting to a trunk port, sending out BPDUs, whereas this particular Catalyst has the ports set as access ports. When the Catalyst receives the BPDUs, an error is generated and the ports are placed in an *inconsistent* state:

```
%SPANTREE-7-RECV_1Q_NON_
TRUNK: Received 802.1Q BPDU on non trunk FastEthernet0/1 VLAN1.
%SPANTREE-7-BLOCK_PORT_
TYPE: Blocking FastEthernet0/1 on VLAN0001. Inconsistent port type.
```

Before the aggregated links are created, this issue needs to be resolved. Therefore on the Catalyst, in global configuration mode, the following is entered:

```
interface range fa0/1 - 3
shut
switchport mode trunk
no shut
```

The ports should now be up and by entering the `show spanning-tree interface` command on the EX switch it is possible to tell which ports are actually forwarding traffic and which are blocking traffic.

```
root@EXSWITCH> show spanning-tree interface

Spanning Tree interface parameters for instance 0

Interface    Port ID  Designated Designated       Port    State  Role
                      port ID    bridge ID            Cost
ge-0/0/12.0  128:525  128:1      32769.001647a7bbc0    200000  FWD    ROOT
ge-0/0/13.0  128:526  128:2      32769.001647a7bbc0    200000  BLK    ALT
ge-0/0/14.0  128:527  128:3      32769.001647a7bbc0    200000  BLK    ALT
```

Now that connectivity has been verified, configuration for the aggregated link can begin. The second step is to tell the EX Switch how many aggregated links will exist on this switch. In this case, it will be one, therefore the following is entered:

```
set chassis aggregated-devices ethernet device-count 1
```

The third step is to add each physical port to the logical port. At Juniper, the logical ports begin with the letters *AE,* and in this case, port 0 will be used as this is the first aggregated link. Therefore physical ports ge-0/0/12, ge-0/0/13, and ge-0/0/14 will be added to logical port ae0. The last entry in this third configuration step tells the switch to use LACP as the protocol (Note that LACP is configured as active on the EX switch. When running LACP at least one switch must be set as active for the protocol to function.)

```
set interfaces ge-0/0/12 ether-options 802.3ad ae0
set interfaces ge-0/0/13 ether-options 802.3ad ae0
set interfaces ge-0/0/14 ether-options 802.3ad ae0
set interfaces ae0 aggregated-ether-options lacp active
```

Now would be a good time to check whether the configuration is correct. Using the `commit check` command, it is clear that there is a problem:

```
root@EXSWITCH# commit check
[edit interfaces ge-0/0/12]
  'unit 0'
     logical unit is not allowed on aggregated links
error: configuration check-out failed

ge-0/0/12 {
    ether-options {
        802.3ad ae0;
    }
    unit 0 {
        family ethernet-switching;
```

You can see that under each physical interface configuration is the setting to make the port a standard enternet port, and that will conflict with the aggregated link. Therefore let's delete that from the configuration:

```
delete interfaces ge-0/0/12 unit 0
delete interfaces ge-0/0/13 unit 0
delete interfaces ge-0/0/14 unit 0

root@EXSWITCH# commit check
configuration check succeeds
```

Moving over to the Catalyst, entering the following is the equivilent of the `ether-options 802.3ad ae0` as entered on the EX. One difference here is that this will automaticaly create the logical interface. On the Catalyst this interface is named *po1*.

```
interface range fa0/1 - 3
channel-protocol lacp
channel-group 1 mode active
```

Now the aggregated links have been created. After commiting the changes, using `run show lacp interfaces` while still in configuration mode should indicate the status of the aggregated link. In this instance, running this configuration returns no entries listed, meaning the link is non existent:

```
root@EXSWITCH# run show lacp interfaces

{master:0}[edit]
root@EXSWITCH#
```

By using the `run show interfaces terse ae0` command, it is clear that the link is down even though administratively it is up:

```
root@EXSWITCH# run show interfaces terse ae0
Interface               Admin Link Proto    Local                 Remote
ae0                     up    down
```

Looking at the log file *messages*, the issue becomes clear: the no *link-speed* option has been set.:

```
EXSWITCH dcd[37154]: ae0 : Warning: aggregated-ether-options link-
speed no kernel value! default to  0
```

The interesting part about this scenario is that the Catalyst has only fast Ethernet ports, whereas the EX switch has gigabit Ethernet. The EX switch has to be told that the port speed will be 100 megabit and this needs to be done on the logical port and the physical ports, too:

```
set interfaces ae0 aggregated-ether-options link-speed 100m
set interfaces ge-0/0/12 ether-options speed 100m
set interfaces ge-0/0/13 ether-options speed 100m
set interfaces ge-0/0/14 ether-options speed 100m
```

Now issue `run show lacp interfaces` again, and the results should show that the links are up, however, the *mux state* is currently showing as detached:

```
root@EXSWITCH# run show lacp interfaces
Aggregated interface: ae0
    LACP state:    Role   Exp  Def  Dist  Col  Syn  Aggr  Timeout  Activity
      ge-0/0/12   Actor    No  Yes    No   No   No   Yes     Fast    Active
      ge-0/0/12 Partner    No  Yes    No   No   No   Yes     Fast   Passive
      ge-0/0/13   Actor    No  Yes    No   No   No   Yes     Fast    Active
      ge-0/0/13 Partner    No  Yes    No   No   No   Yes     Fast   Passive
      ge-0/0/14   Actor    No  Yes    No   No   No   Yes     Fast    Active
      ge-0/0/14 Partner    No  Yes    No   No   No   Yes     Fast   Passive
    LACP protocol:        Receive State  Transmit State          Mux State
      ge-0/0/12               Defaulted   Fast periodic           Detached
      ge-0/0/13               Defaulted   Fast periodic           Detached
      ge-0/0/14               Defaulted   Fast periodic           Detached
```

The simple reason is that although the links have been created, the port ae0 was created with no configuration and this is the same issue faced when the two switches were originally connected. The aggregated port on the EX switch needs to be set as a trunk port:

```
set interfaces ae0 unit 0 family ethernet-switching port-mode trunk
```

If the `show lacp interfaces` command is run now, it should report that the *mux state* is now *collecting* and *distributing*, which means full connectivity.

```
root@EXSWITCH> show lacp interfaces
Aggregated interface: ae0
    LACP state:   Role    Exp  Def  Dist Col  Syn  Aggr  Timeout  Activity
      ge-0/0/12   Actor    No   No   Yes Yes  Yes   Yes     Fast    Active
      ge-0/0/12 Partner    No   No   Yes Yes  Yes   Yes     Slow    Active
      ge-0/0/13   Actor    No   No   Yes Yes  Yes   Yes     Fast    Active
      ge-0/0/13 Partner    No   No   Yes Yes  Yes   Yes     Slow    Active
      ge-0/0/14   Actor    No   No   Yes Yes  Yes   Yes     Fast    Active
      ge-0/0/14 Partner    No   No   Yes Yes  Yes   Yes     Slow    Active
    LACP protocol:   Receive State  Transmit State          Mux State
      ge-0/0/12            Current   Slow periodic Collecting distributing
      ge-0/0/13            Current   Slow periodic Collecting distributing
      ge-0/0/14            Current   Slow periodic Collecting distributing
```

Should these switches be left as they are currently configured, they would work fine and begin forwarding traffic. But according to the brief at the beginning of this recipe, the link should only be forwarding traffic from VLAN 600. As these switches were set as default, VLAN 600 also needs to be created, and then the link needs to be told to allow VLAN 600 only:

```
set vlans VLAN600 vlan-id 600
set interfaces ae0 unit 0 family ethernet-switching port-mode trunk
set interfaces ae0 unit 0 family ethernet-switching vlan members VLAN600
```

To perform the same on the Catalyst, the following configuration is entered:

```
vlan 600
exit
interface range fa0/1 - 3
switchport trunk allowed vlan 600
```

Giving the VLAN interface an IP address isn't a necessary step unless the switch is performing inter-VLAN routing. In this case, the switches will be, therefore an address is added to the interface `vlan.600` on the EX switch, then VLAN600 is assigned to the `vlan.600` interface to bind them together:

```
set interfaces vlan unit 600 family inet address 10.10.10.1/24
set vlans VLAN600 l3-interface vlan.600
```

There is no such need to bind the VLAN to an interface on the Catalyst as it does it for you, so all that is necessary is to add an IP address to the interface VLAN600:

```
interface Vlan600
 ip address 10.10.10.2 255.255.255.0
 no shut
```

Let's issue a ping from the EX switch to confirm whether each switch can reach the other successfully.

```
root@EXSWITCH> ping 10.10.10.2
PING 10.10.10.2 (10.10.10.2): 56 data bytes
64 bytes from 10.10.10.2: icmp_seq=0 ttl=255 time=37.901 ms
64 bytes from 10.10.10.2: icmp_seq=1 ttl=255 time=3.295 ms
64 bytes from 10.10.10.2: icmp_seq=2 ttl=255 time=4.222 ms
64 bytes from 10.10.10.2: icmp_seq=3 ttl=255 time=3.373 ms
```

The configurations shown previously were for Layer 2 aggregated links, meaning these are standard Ethernet ports that participate in Spanning Tree. Aggregated links can also be set as a Layer 3 ports, meaning you can assign an IP address, and the port behaves as if it was the port of a router. Converting the ports from L2 to L3 on the EX switch requires removing `ethernet-switching` under `interfaces <interface-name> unit 0` and replacing it with `inet address <ip address>`:

```
delete interfaces ae0 unit 0 family ethernet-switching
set interfaces ae0 unit 0 family inet address 10.10.10.1/24
```

To make an L2 port an L3 port or *routed port* on the Catalyst, it is necessary to run the `no switchport` command under interface configuration, then an IP address is assigned to the interface po1:

```
interface range fa0/1 - 3
no switchport
interface po1
no switchport
ip address 10.10.10.2 255.255.255.0
```

Finally, issuing a ping from the Catalyst to the EX switch should receive a response, therefore confirming connectivity.

```
CATALYST#ping 10.10.10.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.10.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms
```

Recipe 2

Using Junos Space Security Director to Create IPsec VPNs

Problem

IPsec VPNs are very widely used today and they have been for quite a while – it's typically a cheaper way of establishing secure communications versus leased lines, especially when you need to connect globally. In most cases, you have probably configured your IPsec VPNs via the command line, as with so many other features. But as your network grows, you might find yourself relying on scripts and other methods to deploy your IPsec configurations to the masses, which can be tedious to manage. Let's use Junos Space instead.

Solution

We'll use Junos Space to create and manage IPsec VPNs from Space's GUI, versus a command line interface.

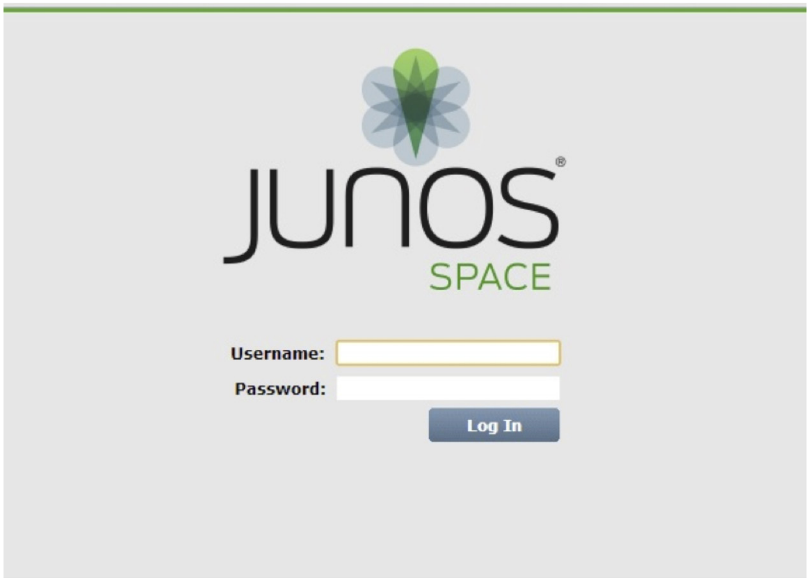
Junos Space is a platform that provides a single interface to manage all of your network devices, and specifically at Security Director, which is used to manage all of your SRX security policies, IDS, IPsec VPNs, and more.

For this recipe's example, let's create an IPsec VPN between two SRX devices. One will be our headquarters, or *HQ*, and the other will be a vendor, let's all it *ABC Corp*. There will be no need for NAT, so a *policy-based* IPsec VPN will be used instead of a *route-based* IPsec VPN.

MORE? See also the recipe in this *Ambassadors' Cookbook*: *SRX to ASA Policy Based IPSec VPN*, for another policy-based IPsec VPN.

Step 1: Log In

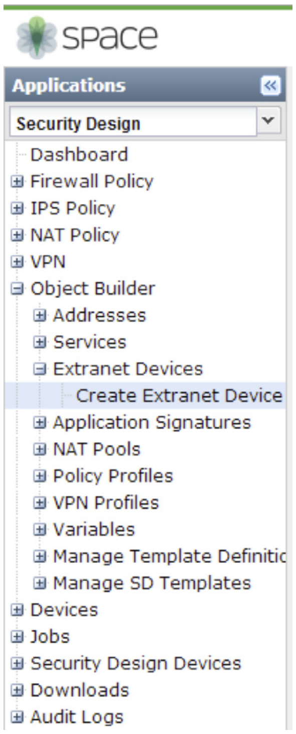
First, let's start by logging in to Junos Space. Open up a web browser and enter the URL of your Junos Space server:



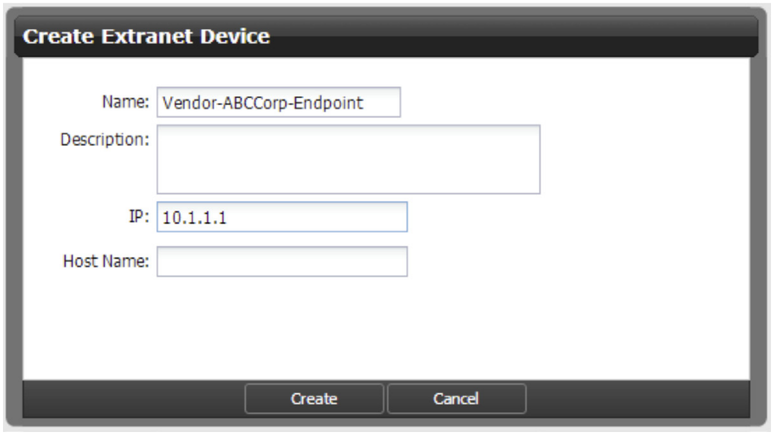
Step 2: Create Extranet Device

Once you are logged in, select Security Design from the Applications menu. You will have to add an entry for our vendor SRX under the Extranet Devices section by clicking on Object Builder > Extranet Devices > Create Extranet Device.

NOTE Extranet Devices are firewalls for which Junos Space does not directly control and manage all the settings. Extranet device objects allow Junos Space to reference a third-party device in a configuration even though you do not have a log in or other control of the device.



Enter a name for the vendor's SRX, as well as their Internet facing (public) IP address.



Name: Vendor-ABCCorp-Endpoint

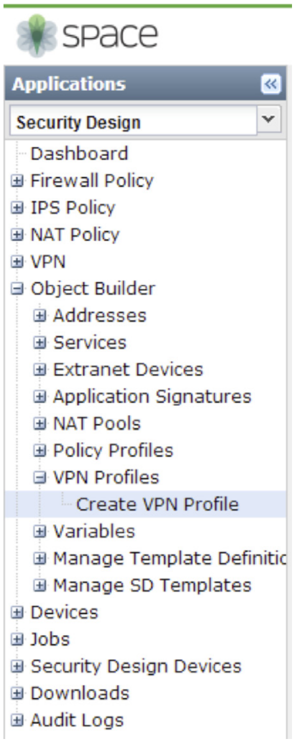
Description:

IP: 10.1.1.1

Host Name:

Step 3: Create a VPN Profile

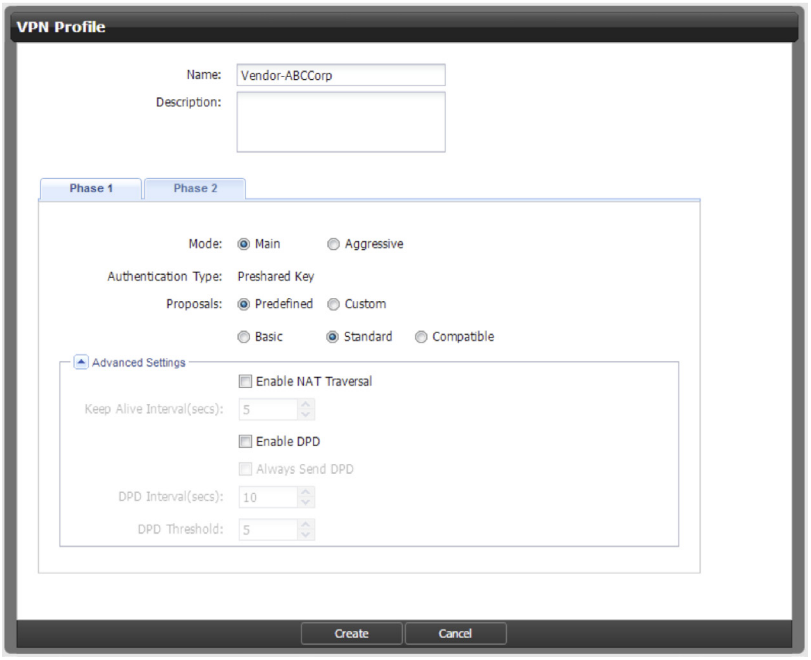
Now you will need to create a VPN policy for the vendor. This will include any and all Phase 1 and Phase 2 policies. To do this, use the navigation menu on the left, and select Object Builder > VPN Profiles > Create VPN Profile.



Now let's start filling in the details for our vendor VPN Policy:

- Name: "Vendor-ABCCorp"
- Description: Optional, so we'll just leave it blank for now.
- Mode: Main
- Proposals: Predefined > Standard
- For our Phase 1 information: You'll be using Main mode, and a pre-defined set of proposal sets. "Standard" will give you the following proposals:
  - Proposal 1: Preshared key, 3DES encryption, and Diffie-Hellman Group 2 and SHA-1 authentication.
  - Proposal 2: Preshared key, Advanced Encryption Standard (AES) 128-bit encryption, and Diffie-Hellman Group 2 and SHA-1 authentication.

Once finished, it should look like this:


VPN Profile

Name: Vendor-ABCCorp

Description:

Phase 1 | Phase 2

Mode: [✓ Main] Aggressive

Authentication Type: Preshared Key

Proposals: [✓ Predefined] Custom

Basic [✓ Standard] Compatible

Advanced Settings

☐ Enable NAT Traversal

Keep Alive Interval(secs): 5

☐ Enable DPD

☐ Always Send DPD

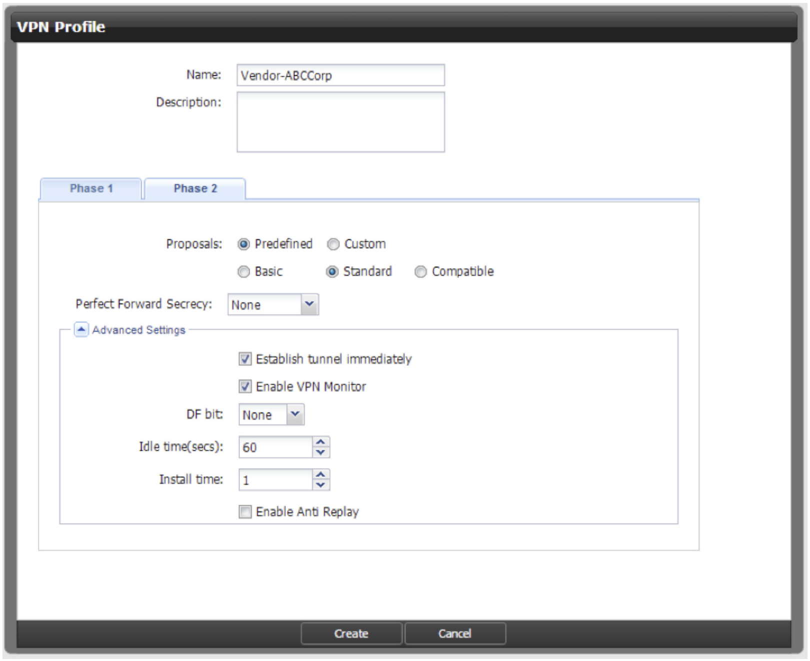
DPD Interval(secs): 10

DPD Threshold: 5

Next, configure the Phase 2 information by clicking on the Phase 2 tab, and then filling in the following information:

- Proposals: Predefined > Standard
- Perfect Forward Secrecy: None
- Under Advanced Settings, enable:
  - Establish tunnel immediately
  - Enable VPN Monitor

Once you have all the settings configured, your Phase 2 should look like the following:


VPN Profile

Name: Vendor-ABCCorp

Description:

Phase 1 | Phase 2

Proposals: (● Predefined) ○ Custom

○ Basic (● Standard) ○ Compatible

Perfect Forward Secrecy: None

Advanced Settings

☑ Establish tunnel immediately

☑ Enable VPN Monitor

DF bit: None

Idle time(secs): 60

Install time: 1

☐ Enable Anti Replay

Note that the Enable Anti Replay is not checked – and that by default it is enabled. Once you have everything configured the way you want it, click on the Create button to create your VPN Profile.

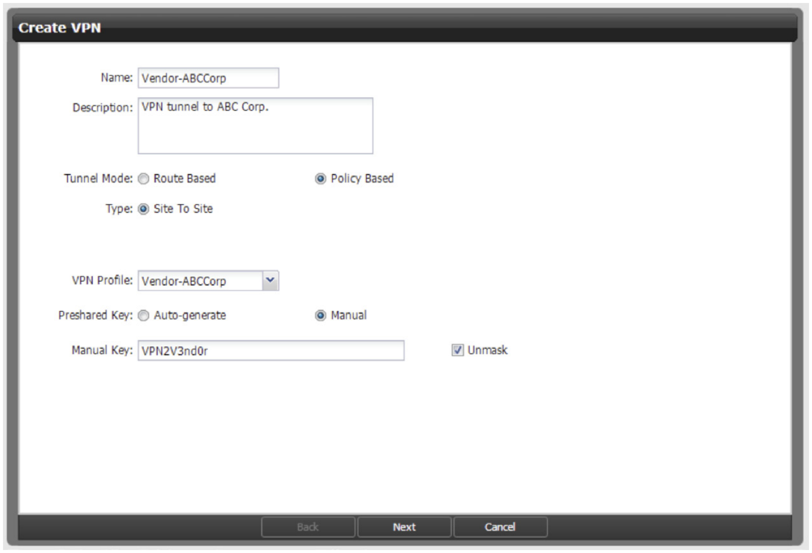
Step 4: Create the IPsec VPN Tunnel

Now that you have created the Extranet Device for your vendor, as well as their VPN Profile, it's time to create our IPsec tunnel! Again, using the left-hand menu, navigate to VPN > Create VPN.

On the Create VPN screen, fill in the details for the IPsec connection to your vendor using the following:

- Name: "Vendor-ABCCorp"
- Description: You can put whatever you'd like here, and it is optional. Let's use "VPN tunnel to ABC Corp."
- Tunnel Mode: You'll be using Policy Based, since both devices are SRX's, and you will *not* be using NAT.
- Type: Site-to-Site
- VPN Profile: Select the vendor's VPN Profile that you created earlier: "Vendor-ABCCorp"
- Preshared Key: You'll be creating your own secret key, but here let's use: *VPN2V3nd0r*

Once finished, your IPsec VPN settings should look like the following:



Create VPN

Name: Vendor-ABCCorp

Description: VPN tunnel to ABC Corp.

Tunnel Mode: ○ Route Based [✓ Policy Based]

Type: [✓ Site To Site]

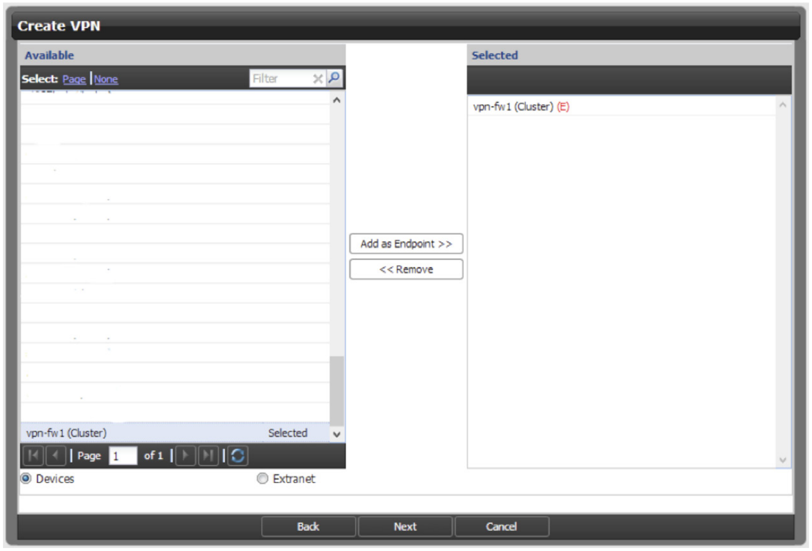
VPN Profile: Vendor-ABCCorp

Preshared Key: ○ Auto-generate [✓ Manual]

Manual Key: VPN2V3nd0r [✓ Unmask]

Click the Next button to proceed.

Now, you need to select the two SRX's with the tunnel between them. Select the HQ SRX first. Make sure the Devices radio button at the bottom is checked, select your VPN SRX (*vpn-fw1* in this example), click *Add as Endpoint*, and then click *Next* when you are finished.



Create VPN

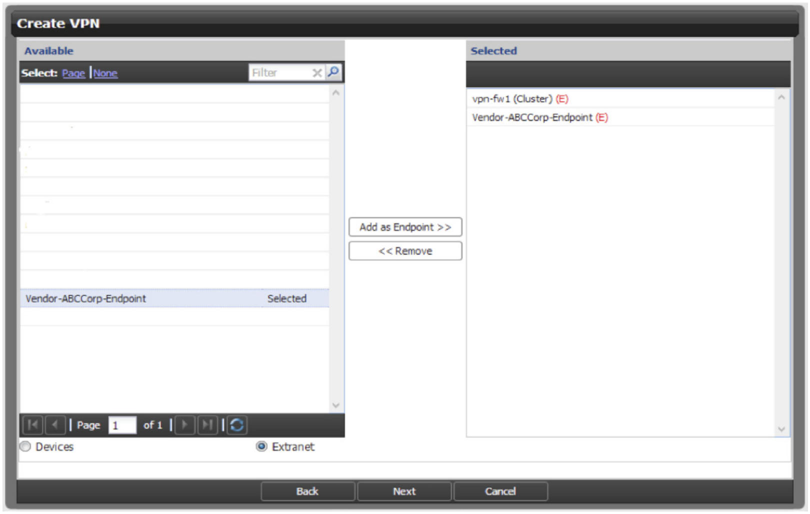
Available: vpn-fw1 (Cluster) Selected

Selected: vpn-fw1 (Cluster) (E)

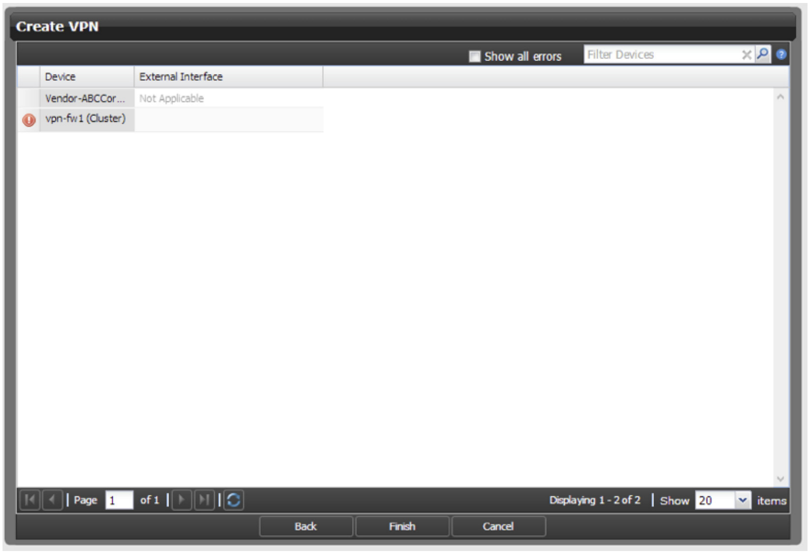
[✓ Devices] ○ Extranet

Next, we'll be selecting our vendor's SRX. Change the radio button at the bottom to *Extranet*, and find the entry for your vendor *Vendor-ABCCorp-Endpoint*.

One you have both of your endpoints selected, click the Next button at the bottom to proceed.



On the next Create VPN screen, you need to select the external interface (public) of your SRX. Click on the box next to your firewall (vpn-fw1) and select your Internet facing IP, as shown here:



Device	External Interface
Vendor-ABCCor...	Not Applicable
vpn-fw1 (Cluster)	

Since Junos Space does not manage our vendor's SRX, we will be leaving their *External Interface* blank.

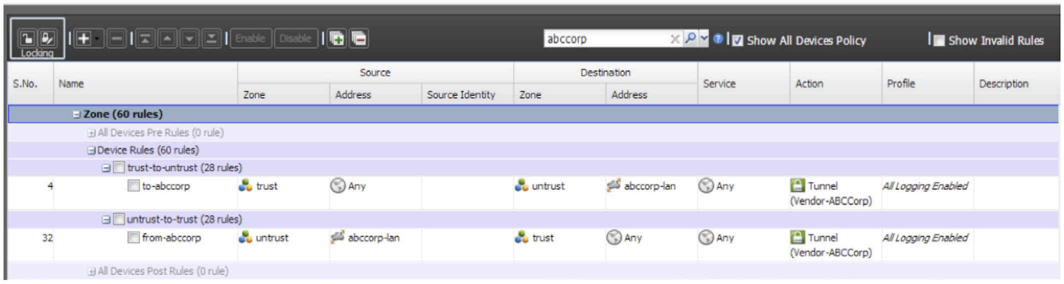
Once you have selected the appropriate interface, click the Finish button.

Step 5: Create Security Policies

Now that the IPsec VPN is set up, you need to configure policies to match the traffic that you want flowing across your tunnel.

Using the left-hand menu, click on Firewall Policy, and select the policy for your SRX here: *vpn-fw1*.

You only need two rules: one from your HQ's to your vendor's protected network, and vice-versa.



S.No.	Name	Source			Destination		Service	Action	Profile	Description
		Zone	Address	Source Identity	Zone	Address				
	Zone (60 rules)									
	All Devices Pre Rules (0 rule)									
	Device Rules (60 rules)									
	trust-to-untrust (28 rules)									
4	to-abccorp	trust	Any		untrust	abccorp-lan	Any	Tunnel (Vendor-ABCCorp)	*All Logging Enabled*	
	untrust-to-trust (28 rules)									
32	from-abccorp	untrust	abccorp-lan		trust	Any	Any	Tunnel (Vendor-ABCCorp)	*All Logging Enabled*	
	All Devices Post Rules (0 rule)									

When creating the security policies for policy-based IPsec VPNs, instead of using the default *permit* or *deny*, you want to select the VPN tunnel that you want the traffic to flow through. This can be done by clicking on the box in the Action field (typically this is permit or deny), and selecting the *Tunnel* option, and choosing the IPsec VPN tunnel that you created previously.

Testing

Now let's test the IPsec VPN tunnel to the vendor, and make sure traffic is flowing through it. To do this, you need to log in to the HQ SRX via the command-line, and enter in the `show security ike security-associations` command and you should get output similar to the following:

```
root@vpn-fw1> show security ike security-associations
--------------------------------------------------------------------------
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
3953325 UP     5324bcab386c9f1e  fc335ffaa7e0fa2c  Main           10.1.1.1
```

As you can see, the Phase 1 state of our VPN tunnel looks good. Let's make sure that Phase 2 is established as well, by using the `show security ipsec security-associations` command:

```
root@vpn-fw1> show security ipsec security-associations
---------------------------------------------------------------------------
  Total active tunnels: 26
  ID    Algorithm       SPI      Life:sec/kb  Mon vsys Port  Gateway
  <34   ESP:3des/md5    36d7a505 28765/unlim   U   root 500   10.1.1.1
  >34   ESP:3des/md5    1548a6e4 28765/unlim   U   root 500   10.1.1.1
```

And everything here is looking good for Phase 2 as well. There is two-way communication between HQ and the vendor.

Summary

As you can see, creating an IPsec VPN tunnel using Junos Space is, in its simplicity, only about a five-step process. This should get you familiar with creating and managing IPsec VPNs within Junos Space as well as all the other options in the program.

Recipe 3

Using Snapshot Images for Mass Deployment

Snapshots are a great feature within Junos. Think of them as a template, if you will. You can configure a basic set of features, take a "snapshot" of the entire device, and then save it on a USB thumb drive to boot off of, or copy to, a new device. This function is available on any device running Junos.

Problem

You have a new project that requires you to install Juniper firewalls at many locations, and the base configuration is the same with minor, location-dependent changes. How can you rapidly deploy a standard configuration to them all without having to manually configure each one?

Solution

Enter the Junos snapshot feature, allowing you to take a "snapshot" of the device and its entire configuration, and then save that snapshot to external media. You can then use that image to copy to subsequent devices so that all of them can be quickly deployed with a standard configuration of your choice.

Creating the Snapshot Image

Let's get started with configuring a basic set of features that we want all of our devices to have. This will include setting the root password, hostname, deleting some of the unwanted default configuration, and enabling DHCP on one of our interfaces:

```
set system root-authentication plain-text-password
delete vlans
delete interfaces vlan
delete security zones security-zone trust interfaces
delete security zones security-zone untrust interfaces
delete interfaces
delete system services web-management
delete system services dhcp
delete system name-server
delete system services telnet
delete system services xnm-clear-text
delete protocols stp
delete security nat
delete system autoinstallation
set system host-name srx1
set interfaces ge-0/0/0 unit 0 family inet dhcp
set security zone security-zone trust host-inbound-traffic system-services all
set security zone security-zone trust host-inbound-traffic protocols all
set security zone security-zone trust interfaces ge-0/0/0.0
```

Once you have configured the device to your liking, go ahead and commit the configuration and then exit configure mode:

```
[edit]
root@srx1# commit and-quit
commit complete
Exiting configuration mode

root@srx1>
```

Now comes the process of creating our image. First, insert a USB thumb drive into port 0. Once you do that, enter in the following:

```
root@srx1> request system snapshot partition media usb
```

This will format the USB drive, create the necessary partitions, and copy everything that is currently on the device hosting the USB drive. It might take a few minutes to complete, and your output will look similar to the following:

```
Clearing current label...
Partitioning usb media (/dev/da1) ...
Partitions on snapshot:

  Partition  Mountpoint  Size    Snapshot argument
      s1a    /altroot    2.4G    none
      s2a    /           2.4G    none
      s3e    /config     185M    none
      s3f    /var        2.1G    none
      s4a    /recovery/software 216M none
      s4e    /recovery/state 15M none
Copying '/dev/da0s1a' to '/dev/da1s1a' .. (this may take a few minutes)
Copying '/dev/da0s2a' to '/dev/da1s2a' .. (this may take a few minutes)
Copying '/dev/da0s3e' to '/dev/da1s3e' .. (this may take a few minutes)
Copying '/dev/da0s3f' to '/dev/da1s3f' .. (this may take a few minutes)
Copying '/dev/da0s4e' to '/dev/da1s4e' .. (this may take a few minutes)
Copying '/dev/da0s4a' to '/dev/da1s4a' .. (this may take a few minutes)
The following filesystems were archived: /altroot / /config /var /recovery/state /
recovery/software
```

Once this process has completed, and you are returned to your prompt, you can now remove the USB drive.

Deploying the Snapshot Image to New Devices

Now let's use our newly created image on a new device that hasn't been configured yet. Once you have powered on your new device, and it has booted up, log in with the default credentials (*root*, no password), then enter in the following to get to the operational mode prompt:

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

Insert your USB drive with the image you created earlier into the new device's USB port 0. Once you have done that, enter the following command:

```
root> request system reboot media usb
```

When Junos asks you to reboot the system, choose "yes."

```
Reboot the system ? [yes,no] (no) yes
```

It's essentially telling the device to boot off of the image stored on the USB drive. Once it has booted up, log in using the credentials that you supplied when you initally configured the image on the first device, and enter operational mode.

Now that you are running off of the image, you need to copy it from the USB drive onto the internal hard drive. Do this by entering the following command at the operational mode prompt:

```
root@srx1> request system snapshot media internal partition
```

This will start the image copy process and might take a few minutes to complete. When it has finished, you should see output similar to:

```
Clearing current label...
Partitioning internal media (/dev/da0) ...
Partitions on snapshot:
    Partition  Mountpoint  Size    Snapshot argument
        s1a    /altroot    610M    none
        s2a    /           617M    none
        s3e    /config     46M     none
        s3f    /var        618M    none
        s4a    /recovery/software 56M none
        s4e    /recovery/state 5.7M none
Copying '/dev/da1s1a' to '/dev/da0s1a' .. (this may take a few minutes)
Copying '/dev/da1s2a' to '/dev/da0s2a' .. (this may take a few minutes)
Copying '/dev/da1s3e' to '/dev/da0s3e' .. (this may take a few minutes)
Copying '/dev/da1s3f' to '/dev/da0s3f' .. (this may take a few minutes)
Copying '/dev/da1s4e' to '/dev/da0s4e' .. (this may take a few minutes)
Copying '/dev/da1s4a' to '/dev/da0s4a' .. (this may take a few minutes)
The following filesystems were archived: /altroot / /config /var /recovery/state /
recovery/software
```

Once the image is copied onto the device, you can now boot off of the internal storage, the normal way the device does. Since you are currently booted-off of the USB drive, you need to tell the device to boot off of internal storage:

```
root@srx1> request system reboot media internal
```

Once the device has booted up, you can now log in and verify that the configuration is exactly the same as when you initially configured the image.

You can now *safely* remove the USB drive from your device.

And as you can see, in just a few steps you were able to quickly set up a new device with a default configuration that you specified – a huge help in mass deployments, where a lot of the configuration will be similar across all of your devices.

TIP It is also handy to keep a snapshot as a sort of *Gold* configuration, if you will, so that if you ever need to get back to a standard, you can quickly do so.

Recipe 4

Redundant VPN with Fast Failover

Problem

Due to its popularity, there's a growing need for redundancy in IPsec VPN type connections. For instance, a failover should ideally occur within a second when a VPN is carrying VoIP or Video and the primary link is lost. Junos has good solutions for VPN redundancy and some of them will be discussed here, but the focus of this solution is on redundancy, and details or explanations on how to configure VPNs can be found elsewhere, either online, at Juniper Techpubs (www.juniper.net/techpubs), or in the *Day One* library of books (www.juniper.net/dayone).

Solution

Let's set the stage with an example where two networks are connected over two separate ISPs. ISP1 is primary, and ISP2 is used as a backup.

```
Junos version tested on:      12.1X44-D10.4
```

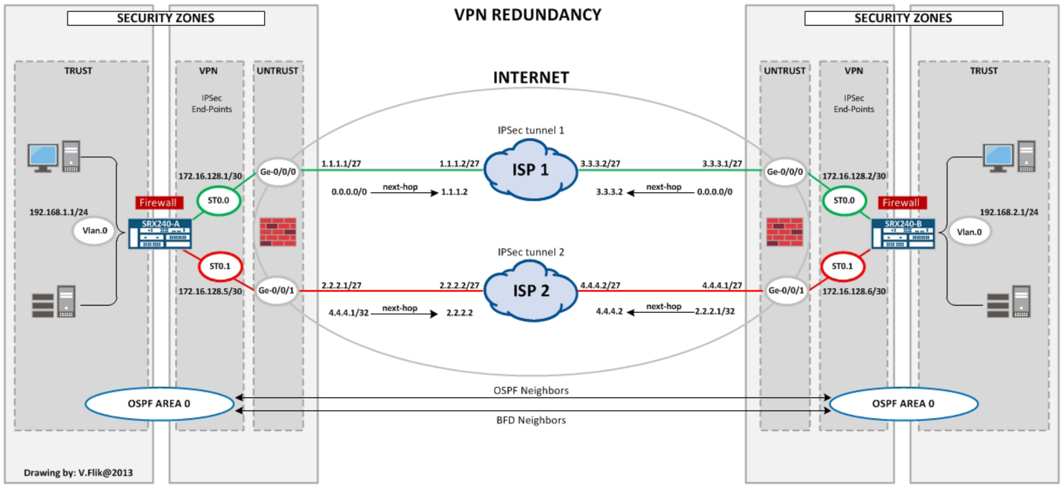


Figure 4.1 This Recipe's Topology

The basic VPN configuration for this recipe's setup is here:

SRX-1 (under security)

```
ike {
    proposal aes128-sha2-dh5 {
        authentication-method pre-shared-keys;
        dh-group group5;
        authentication-algorithm sha-256;
        encryption-algorithm aes-256-cbc;
        lifetime-seconds 28800;
    }
    policy srx1-srx2-policy {
        mode main;
        proposals aes128-sha2-dh5;
        pre-shared-key ascii-text "$9$dOw2oDi.z39JG39ApREdbs2JGjHqfQF"; ## SECRET-
DATA
    }
    gateway srx-2-over-ISP1 {
        ike-policy srx1-srx2-policy;
        address 3.3.3.1;
        external-interface ge-0/0/0.0;
    }
    gateway srx-2-over-ISP2 {
        ike-policy srx1-srx2-policy;
        address 4.4.4.1;
        external-interface ge-0/0/1;
    }
}
ipsec {
    vpn-monitor-options {
        interval 2;
        threshold 3;
    }
    proposal esp-aes128-sha2 {
        protocol esp;
        encryption-algorithm aes-256-cbc;
        lifetime-seconds 3600;
    }
    policy ipsec-srx1-srx2-policy {
        perfect-forward-secrecy {
            keys group5;
        }
        proposals esp-aes128-sha2;
    }
    vpn vpn-srx2-over-ISP1 {
        bind-interface st0.0;
        vpn-monitor;
        ike {
            gateway srx-2-over-ISP1;
            ipsec-policy ipsec-srx1-srx2-policy;
        }
        establish-tunnels immediately;
    }
    vpn vpn-srx2-over-ISP2 {
        bind-interface st0.1;
        vpn-monitor;
        ike {
            gateway srx-2-over-ISP2;
            ipsec-policy ipsec-srx1-srx2-policy;
        }
        establish-tunnels immediately;
    }
}
```

SRX-2 (under security)

```
ike {
    proposal aes128-sha2-dh5 {
        authentication-method pre-shared-keys;
        dh-group group5;
        authentication-algorithm sha-256;
        encryption-algorithm aes-256-cbc;
        lifetime-seconds 28800;
    }
    policy srx1-srx2-policy {
        mode main;
        proposals aes128-sha2-dh5;
        pre-shared-key ascii-text "$9$dOw2oDi.z39JG39ApREdbs2JGjHqfQF"; ## SECRET-
DATA
    }
    gateway srx-1-over-ISP1 {
        ike-policy srx1-srx2-policy;
        address 1.1.1.1;
        external-interface ge-0/0/0.0;
    }
    gateway srx-1-over-ISP2 {
        ike-policy srx1-srx2-policy;
        address 2.2.2.1;
        external-interface ge-0/0/1;
    }
}
ipsec {
    vpn-monitor-options {
        interval 2;
        threshold 3;
    }
    proposal esp-aes128-sha2 {
        protocol esp;
        encryption-algorithm aes-256-cbc;
        lifetime-seconds 3600;
    }
    policy ipsec-srx1-srx2-policy {
        perfect-forward-secrecy {
            keys group5;
        }
        proposals esp-aes128-sha2;
    }
    vpn vpn-srx1-over-ISP1 {
        bind-interface st0.0;
        vpn-monitor;
        ike {
            gateway srx-1-over-ISP1;
            ipsec-policy ipsec-srx1-srx2-policy;
        }
        establish-tunnels immediately;
    }
    vpn vpn-srx1-over-ISP2 {
        bind-interface st0.1;
        vpn-monitor;
        ike {
            gateway srx-1-over-ISP2;
            ipsec-policy ipsec-srx1-srx2-policy;
        }
        establish-tunnels immediately;
    }
}
```

On both SRX devices the tunnel interfaces (st0.0 and st0.1) are placed in a zone called *vpn,* and security policies allow traffic from VPN to trust and vice-versa. Of course, host-inbound-traffic system-service IKE is allowed on the untrust zone:

```
root@SRX-1# show security zones security-zone vpn
host-inbound-traffic {
    system-services {
        traceroute;
        ping;
    }
}
```

```
interfaces {
    st0.0;
    st0.1;
}
root@SRX-1# show security policies from-zone trust to-zone vpn
policy trust-vpn {
    match {
        source-address any;
        destination-address any;
        application any;
    }
    then {
        permit;
    }
}
root@SRX-1# show security policies from-zone vpn to-zone trust
policy vpn-trust {
    match {
        source-address any;
        destination-address any;
        application any;
    }
    then {
        permit;
    }
}
```

To make sure the IKE traffic follows two different ISPs, a static routing entry for the outside of the tunnel is added to the config, overwriting the IKE destination with a /32 route:

```
root@SRX-1# show routing-options
static {
    route 0.0.0.0/0 next-hop 1.1.1.2;
    route 4.4.4.1/32 next-hop 2.2.2.2;
}
```

Of course on SRX-2, 4.4.42 is used as next-hop to reach 2.2.2.1, the outgoing IKE address of SRX-1.

Take a good look at the IPsec configuration of both devices. You will see that ipsec-monitoring is set to a threshold of 3 and an interval of 2 seconds. This results in the interface going to the down state when the VPN is down in 6 seconds (threshold multiplied by interval). This behavior is used to set up the following static routing on the devices:

SRX-1

```
root@SRX-1# show routing-options
static {
    route 0.0.0.0/0 next-hop 1.1.1.2;
    route 4.4.4.1/32 next-hop 2.2.2.2;
    route 192.168.2.0/24 {
        next-hop st0.0;
        qualified-next-hop st0.1 {
            metric 10;
        }
    }
}
```

SRX-2

```
root@srx-2# show routing-options
static {
    route 0.0.0.0/0 next-hop 3.3.3.2;
    route 2.2.2.1/32 next-hop 4.4.4.2;
    route 192.168.1.0/24 {
        next-hop st0.0;
        qualified-next-hop st0.1 {
            metric 10;
        }
    }
}
```

The route over st0.0 is preferred, so the traffic is sent over VPN1 when this VPN is up. When the VPN monitor notices the VPN is down, the route becomes inactive, and the next route (using VPN2) becomes active. Let's take a look at the routing table to see this happening:

```
root@SRX-1> show route 192.168.2/24
inet.0: 13 destinations, 14 routes (13 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
192.168.2.0/24     *[Static/5] 00:00:11
                    > via st0.0
                    [Static/5] 00:00:11, metric 10
                    > via st0.1
```

Now, let's start a ping from SRX-1 to SRX-2 and force a problem in ISP1 after four pings:

```
root@SRX-1> ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1): 56 data bytes
64 bytes from 192.168.2.1: icmp_seq=0 ttl=64 time=2.642 ms
64 bytes from 192.168.2.1: icmp_seq=1 ttl=64 time=8.518 ms
64 bytes from 192.168.2.1: icmp_seq=2 ttl=64 time=3.509 ms
64 bytes from 192.168.2.1: icmp_seq=3 ttl=64 time=2.381 ms
64 bytes from 192.168.2.1: icmp_seq=11 ttl=64 time=2.267 ms
64 bytes from 192.168.2.1: icmp_seq=12 ttl=64 time=2.480 ms
```

Now when ISP1 has a major problem (say the patch cable falls out of the Ethernet port, somehow) it missed eight pings before the fallback to the second VPN took over. (Check the sequence number to confirm this fact!) The routing table now shows this (everything is shown from SRX-1 but the same thing is happening on SRX-2 of course).

```
root@SRX-1> show route 192.168.2/24

inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.2.0/24     *[Static/5] 00:05:13, metric 10
                    > via st0.1
```

The solutions with VPN monitoring work, but not within the desired time. Let's look for a faster solution.

Introducing BFD

Bidirectional Forwarding Detection (BFD) is an open standard protocol for link monitoring. It sends a stream of packets from both sides and allows for a threshold of missed packets. When the threshold is reached, a upper level routing protocol is notified of the problem and is triggered to come into action. This upper level protocol can be BGP, OSPF, static routing, or many others. Let's first take a look at static routes, since a lot is already in place in the config to do so.

On some hardware platforms the BFD processing is distributed to the PFE, allowing for an interval of 100ms between the packet. On SRX's, the BFD sessions are handled by the RE so you should be careful with intervals that are too fast and with too many BFD sessions. It's recommended not to go below 300ms for RE-based BFD sessions, so let's be sure not to do that. One thing to change is the static routes. Until now it was only mentioned that the tunnel interface was next-hop. For BFD you need a local *and* a remote IP address. The local address comes from the interface IP address, and the next-hop is used for the remote address.

The BFD configuration looks like the following:

SRX-1

```
root@srx-2# show routing-options static

route 0.0.0.0/0 next-hop 1.1.1.2;
route 4.4.4.1/32 next-hop 2.2.2.2;
route 192.168.2.0/24 {
    next-hop 172.16.128.2;
    qualified-next-hop 172.16.128.6 {
        metric 10;
    }
    bfd-liveness-detection {
        minimum-interval 300;
    }
}
```

SRX-2

```
root@srx-2# show routing-options static

route 0.0.0.0/0 next-hop 3.3.3.2;
route 2.2.2.1/32 next-hop 4.4.4.2;
route 192.168.1.0/24 {
    next-hop 172.16.128.1;
    qualified-next-hop 172.16.128.5 {
        metric 10;
    }
    bfd-liveness-detection {
        minimum-interval 300;
    }
}
```

BFD is a protocol, so it needs to be added to host-inbound traffic protocol on the VPN zone:

```
root@SRX-1# show security zones security-zone vpn
host-inbound-traffic {
    system-services {
        traceroute;
        ping;
    }
    protocols {
        bfd;
    }
}
```

Now you can check the BFD status with the show BFD session command:

```
root@SRX-1> show bfd session
                                                  Detect   Transmit
Address                  State     Interface      Time     Interval  Multiplier
172.16.128.2             Up        st0.0          0.900     0.300        3
172.16.128.6             Up        st0.1          0.900     0.300        3

2 sessions, 2 clients
Cumulative transmit rate 6.7 pps, cumulative receive rate 6.7 pps
```

Clearly, BFD is up and running! Let's check the routing table and failover once again:

```
root@SRX-1> show route 192.168.2/24

inet.0: 13 destinations, 14 routes (13 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.2.0/24     *[Static/5] 00:15:58
                    > to 172.16.128.2 via st0.0
                    [Static/5] 00:21:12, metric 10
                    > to 172.16.128.6 via st0.1
```

Routing is okay! Now let's ping and check again to see if there's a problem after the fourth packet:

```
root@SRX-1> ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1): 56 data bytes
64 bytes from 192.168.2.1: icmp_seq=0 ttl=64 time=2.532 ms
64 bytes from 192.168.2.1: icmp_seq=1 ttl=64 time=2.397 ms
64 bytes from 192.168.2.1: icmp_seq=2 ttl=64 time=2.389 ms
64 bytes from 192.168.2.1: icmp_seq=3 ttl=64 time=2.343 ms
64 bytes from 192.168.2.1: icmp_seq=5 ttl=64 time=2.277 ms
64 bytes from 192.168.2.1: icmp_seq=6 ttl=64 time=2.384 ms
64 bytes from 192.168.2.1: icmp_seq=7 ttl=64 time=2.308 ms
```

So we only missed one ping ! That's more like it! Using BFD really does the trick.

It might be a good idea to use OSPF instead of static routing – OSPF also supports BFD, so no problem here. You could place the local LAN interface in passive mode and just make the tunnel interfaces active. Using OSPF could save you a lot of static routes, especially on larger VPNs, and BFD assures a fast recovery from broken VPN tunnels. The configuration for basic OSPF would look like this (note OSPF is added as host-inbound-traffic protocol):

SRX-1

```
root@SRX-1# show security zones security-zone vpn
host-inbound-traffic {
    system-services {
        traceroute;
        ping;
    }
    protocols {
        bfd;
        ospf;
    }
}
interfaces {
    st0.0;
    st0.1;
}

root@SRX-1# show protocols ospf
area 0.0.0.0 {
    interface st0.0 {
        metric 10;
        bfd-liveness-detection {
            minimum-interval 300;
        }
    }
    interface st0.1 {
        metric 20;
```

```
            bfd-liveness-detection {
                minimum-interval 300;
            }
        }
        interface vlan.0 {
            passive;
        }
    }
```

SRX-2

```
root@srx-2# show security zones security-zone vpn
host-inbound-traffic {
    system-services {
        traceroute;
    }
    protocols {
        bfd;
        ospf;
    }
}
interfaces {
    st0.0;
    st0.1;
}


root@srx-2# show protocols ospf
area 0.0.0.0 {
    interface st0.0 {
        metric 10;
        bfd-liveness-detection {
            minimum-interval 300;
        }
    }
    interface st0.1 {
        metric 20;
        bfd-liveness-detection {
            minimum-interval 300;
        }
    }
    interface vlan.0 {
        passive;
    }
}
```

NOTE To make sure ISP1 is preferred, make the cost for the link over ISP2 higher.

For the last time, let's take a look at the routing table:

```
root@SRX-1> show route 192.168.2/24

inet.0: 14 destinations, 16 routes (14 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.2.0/24     *[OSPF/10] 00:07:29, metric 11
                    > via st0.0

root@SRX-1> ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1): 56 data bytes
64 bytes from 192.168.2.1: icmp_seq=0 ttl=64 time=2.466 ms
64 bytes from 192.168.2.1: icmp_seq=1 ttl=64 time=2.357 ms
64 bytes from 192.168.2.1: icmp_seq=2 ttl=64 time=2.263 ms
64 bytes from 192.168.2.1: icmp_seq=3 ttl=64 time=2.410 ms
64 bytes from 192.168.2.1: icmp_seq=5 ttl=64 time=2.443 ms
64 bytes from 192.168.2.1: icmp_seq=6 ttl=64 time=2.410 ms
64 bytes from 192.168.2.1: icmp_seq=7 ttl=64 time=2.423 ms
^C
--- 192.168.2.1 ping statistics ---
8 packets transmitted, 7 packets received, 12% packet loss
round-trip min/avg/max/stddev = 2.263/2.396/2.466/0.063 ms
```

So with OSPF only one packet is lost, and there are a lot of other advantages. Finally, let's take a look at the route table in the "fault situation":

```
root@SRX-1> show route 192.168.2.0

inet.0: 12 destinations, 14 routes (12 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.2.0/24     *[OSPF/10] 00:04:55, metric 21
                    > via st0.1
```

As you can see, the metric is 21 where on the primary link it was 11. So our cost manipulation worked.

Summary

The SRX offers a nice way to set up redundancy in VPN links with its support of the open standard BFD protocol.

Recipe 5

Branch VPN Without Local Internet Access

Problem

Branch offices that connect to the corporate network using Internet VPN have direct internet access *and* a secured tunnel to the private corporate network, which can create a split tunnel situation where the default route for the branch uses the Internet and only specified routes traverse the private VPN. A split tunnel can create two difficulties:

- Local Internet access can bypass proxy servers and other security and compliance engines for Internet access on the corporate network.
- The routes used on the VPN tunnel to access various corporate subnets need to be maintained over time.

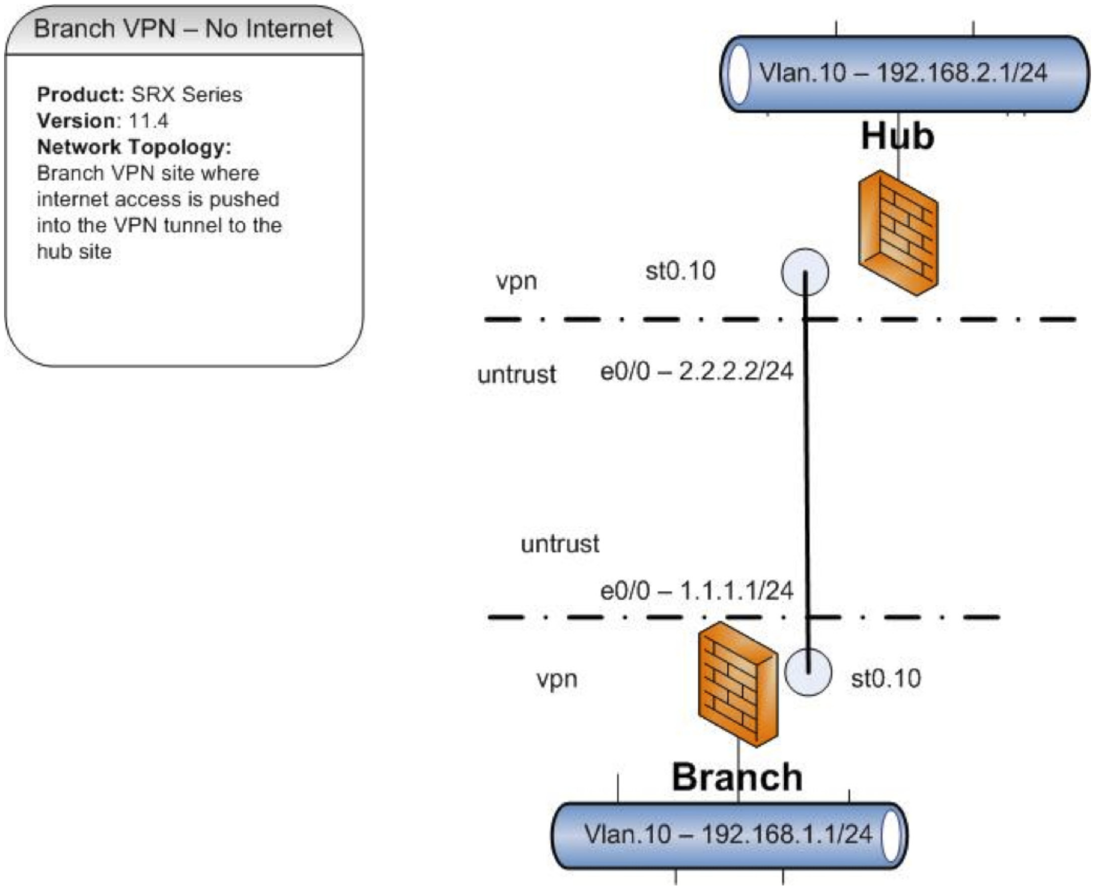
Solution

To solve this predicament, the branch office LAN is configured in a routing instance so that the default route for all traffic on the LAN becomes the VPN tunnel. Ensuring that any and all traffic destined outside the branch LAN resources gets forwarded to the corporate infrastructure and uses the corporate Internet service, security, and compliance configurations. It also ensures that no matter how the corporate network changes over time, traffic from the branch site will be forwarded to the corporate VPN.

Let's go through the four basic steps of the configuration:

- Configure Interfaces for branch LAN and tunnel
- Create a routing instance to separate local LAN and VPN traffic
- Set up zone security
- Set up standard route based VPN

NOTE This recipe was tested on a SRX100 using Junos 11.4r5.5. And note that all configuration statements are shown as they would be entered on the branch device.



Branch VPN – No Internet

Product: SRX Series
Version: 11.4
Network Topology:
Branch VPN site where internet access is pushed into the VPN tunnel to the hub site

Figure 5.1 This Recipe's Topology

Configure Interfaces for branch LAN and tunnel

First put the desired interfaces into the VLAN-VPN and assign an IP range and address:

```
set interfaces fe-0/0/1 unit 0 family ethernet-switching
set interfaces fe-0/0/1 unit 0 vlan members vlan-vpn
set interfaces vlan unit 10 family inet address 192.168.1.1/24
set vlans vlan-vpn vlan-id 10
set vlans vlan-vpn l3-interface vlan.10
```

Now, create the tunnel interface:

```
set interfaces st0 unit 10 family inet
```

Create a routing instance to separate local LAN and VPN traffic

Create the routing instance adding the interfaces and default route for the tunnel:

```
set routing-instances vr-vpn instance-type virtual-router
set routing-instances vr-vpn interface st0.10
set routing-instances vr-vpn interface vlan.10
set routing-instances vr-vpn routing-options static route 0.0.0.0/0 next-hop st0.10
```

Set up zone security

The next configuration places both the tunnel and branch LAN into the same zone so no policies are required for traffic, permitting inbound management traffic on the tunnel. (You could also move one interface to a different zone to require policies to allow traffic instead of this default intrazone relationship.)

```
set  zones security-zone vpn
set  zones security-zone vpn host-inbound-traffic system-services all
set  zones security-zone vpn interfaces vlan.10
set  zones security-zone vpn interfaces st0.10
```

Set up standard route-based VPN

Follow the steps to create the route-based VPN using the tunnel interface created above. Start by configuring the local gateway interface for the VPN in the primary routing instance – which is the public IP address and interface from the local Internet service – and put these into the untrust zone in the primary routing instance:

Configure the Internet static IP address on the gateway interface.

```
set interfaces fe-0/0/0 unit 0 family inet address 1.1.1.2/24
```

Now set the default route for our local Internet service in the primary routing instance:

```
set routing-options static route 0.0.0.0/0 next-hop 1.1.1.1
```

Place this gateway interface into the untrust zone in the primary routing instance.

```
set security zones security-zone untrust interfaces fe-0/0/0.0
```

Now, allow management services on this interface for the VPN ike connections. You can also allow other management services as needed on this interface, but the ike services are required for the VPN connection to establish (since this is an Internet-facing interface, you want to limit the other system services you allow for now).

```
set security zones security-zone untrust host-inbound-traffic system-services ike
```

Next, configure the remote gateway ike parameters for policy and preshared key. Naturally, these parameters must match the configuration at the corporate site where the VPN is established.

```
set security ike policy ike-policy1 mode main
set security ike policy ike-policy1 proposal-set standard
set security ike policy ike-policy1 pre-shared-key ascii-text "sharedsecret"
```

Use this policy set to create the ike gateway.

```
set security ike gateway ike-gate ike-policy ike-policy1
set security ike gateway ike-gate address 2.2.2.2
set security ike gateway ike-gate external-interface fe-0/0/0.0
```

Now create the IPsec parameters for the VPN connection.

```
set security ipsec policy vpn-policy1 proposal-set standard
```

Then use this policy to create the IPsec phase 2 connection and tie this to the tunnel interface created above.

```
set security ipsec vpn ike-vpn ike gateway ike-gate
set security ipsec vpn ike-vpn ike ipsec-policy vpn-policy1
set security ipsec vpn ike-vpn bind-interface st0.10
```

Finally, set the `vpn mtu` to the most common value. This prevents fragmentation of traffic across the tunnel.

```
set security flow tcp-mss ipsec-vpn mss 1350
```

References

Routing Instances Overview:

http://www.juniper.net/techpubs/en_US/junos11.4/topics/concept/routing-instances-overview.html

Understanding IPSEC VPN:

http://www.juniper.net/techpubs/en_US/junos11.4/information-products/topic-collections/security/software-all/security/index.html?topic-54272.html

Configuring Route-Based VPN:

http://www.juniper.net/techpubs/en_US/junos11.4/information-products/topic-collections/security/software-all/security/index.html?topic-52842.html

Virtual Router Support in VPN:

http://www.juniper.net/techpubs/en_US/junos11.4/information-products/topic-collections/security/software-all/security/index.html?topic-51414.html

Understanding Virtual Router Limitations:

http://www.juniper.net/techpubs/en_US/junos11.4/information-products/topic-collections/security/software-all/security/index.html?topic-51413.html

Recipe 6

Configuring OSPF Security

OSPF is a flexible Interior Routing Protocol, and within its many options are ways to make the protocol more secure to prevent unwanted personnel influencing your network policies. The purpose of this recipe is to introduce some of the OSPF security features available in Junos, explain how to enable these features, and show how, once enabled, connectivity to a device running IOS is maintained.

Problem

OSPF is one of the most popular Interior Gateway Protocols in use today because this IGP is an open standard supported by many vendors. Configuring OSPF to build a simple network is relatively straightforward. Unfortunately, however, attackers can use this ease of configuration to compromise your network or deny services to your organization. Thankfully, OSPF also includes security mechanisms to protect your network and including this option in your OSPF configuration just makes good sense. This recipe goes through the types of OSPF security, how to enable them under Junos, and shows you how to establish a secure neighbor relationship with a device running IOS.

WARNING Enabling OSPF security on a live network will briefly render some, if not all subnets, as unreachable, therefore it is considered best practice to add this configuration "out of hours."

Solution

In this recipe's example solution, there are two routers, a J2320 and a router running IOS, connected via a broadcast network across OSPF Area 0.0.0.0 via interfaces ge-0/0/0.0 and fa0/0, respectively, as shown in Figure 6.1. In addition, the J2320 has an interface in Area 0.0.0.2 and the IOS device has an interface in Area 0.0.0.1. No other routers are located in the non-backbone areas.

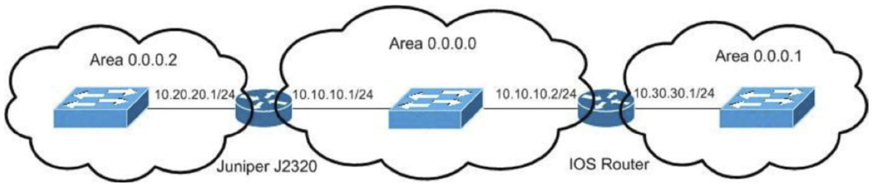


Figure 6.1 This Recipe's Topology

The first task is to enable OSPF on the relevant interfaces, and in IOS, add the appropriate network commands. The configuration would be as follows, first Junos, then IOS:

```
[edit protocols]
root@J2320# show
ospf {
    area 0.0.0.0 {
        interface ge-0/0/0.0;
    }
    area 0.0.0.2 {
        interface ge-0/0/1.0;
    }
}

IOSRouter# sh run | section router ospf
router ospf 1
 log-adjacency-changes
 network 10.10.10.2 0.0.0.0 area 0
 network 10.30.30.1 0.0.0.0 area 1
```

Once OSPF is enabled, you should see the following entry in the log file:

```
rpd[1210]: RPD_OSPF_NBRUP: OSPF neighbor 10.10.10.2 (realm ospf-v2 ge-0/0/0.0 area
0.0.0.0) state changed from Loading to Full due to LoadDone (event reason: OSPF loading
completed)
```

Alternatively, you could use the `show ospf neighbor` command to check the relationship status:

```
root@J2320> show ospf neighbor
Address          Interface              State     ID               Pri  Dead
10.10.10.2       ge-0/0/0.0             Full      10.10.10.2       1    37
```

Once a neighbor relationship has been confirmed, you can begin to enable security.

NOTE RFC2328, OSPF Version 2, allows for three different authentication types between devices running OSPF. The first option is null, meaning none, just as already configured, so let's not discuss the first option much further. The second option is "simple password," meaning the password is sent between devices using plain text. The reality is, it's unlikely you would use a simple password because it's not really a secure security mechanism. The third authentication option is considered a more secure option, as the password is sent in an encrypted format in what is known as a hashing algorithm.

For the purpose of this recipe, let's first use a simple password configuration and then compare it to the authentication option. Begin on the J2320 and enter the following in the top command:

```
set protocols ospf area 0 interface ge-0/0/0.0 authentication simple-password i<3junos
```

The password in this case is *i<3junos*.

If you look at the configuration in Junos, you can see that although the password is sent as clear text, it is still stored in an encrypted format in the active configuration:

```
[edit protocols ospf area 0.0.0.0]
root@J2320# show
interface ge-0/0/0.0 {
    authentication {
        simple-password "$9$42aGi369pO1.PRSlMN-DikPz6"; ## SECRET-DATA
    }
}
```

Once we commit this configuration, you are going to lose the neighbor relationship when the dead timer expires. An error similar to the one shown here will be entered into the messages log file, indicating a lost neighbor:

```
J2320 rpd[1210]: RPD_OSPF_NBRDOWN: OSPF neighbor 10.10.10.2 (realm ospf-v2 ge-0/0/0.0
area 0.0.0.0) state changed from Full to Down due to InActiveTimer (event reason:
neighbor was inactive and declared dead)
```

To resolve this issue you need to enable authentication on the IOS device by typing the following into global configuration mode:

```
router ospf 1
area 0 authentication
int fa0/0
ip ospf authentication-key i<3junos
```

It's important to note that if you look at the running configuration in IOS, the password is stored as plain text by default, but is encrypted once you enter the `service password-encryption` command.

You can confirm that the neighbor relationship has been reestablished by checking the messages log file once more or by using the Junos `show ospf neighbor` command.

The third authentication option is considered a more secure option as the password is sent in an encrypted format. This encryption is performed by running a hashing algorithm called *MD5* against the password. The resulting string of characters, known as a *message digest,* is sent to the neighbor instead of the actual password. In theory, if an attacker managed to retrieve the message digest by sniffing the network, he wouldn't be able to decrypt the message to find out what the real password was.

NOTE Although the password is encrypted, the actual OSPF Link State Advertisements and Requests are still sent in an unencrypted format. Remember, the purpose of the security is to prevent unwanted neighbor relationships, not to hide the advertised subnet information.

Enabling this more secure form of encryption isn't much more complicated than the simple password method, at least, not as far as Junos is concerned. The first step would be to remove the simple password configured previously, done thusly:

```
edit protocols ospf area 0.0.0.0 interface ge-0/0/0.0
delete authentication
```

Next, you can add the MD5 authentication option:

```
set authentication md5 1 key i<3junos
commit
```

At this point you should lose connectivity with the IOS neighbor, so you need to remove the simple password authentication on the IOS device:

```
Int fa0/0
no ip ospf authentication-key
router ospf 1
no area 0 authentication
```

Then you need to add MD5 authentication:

```
area 0 authentication message-digest
int fa0/0
ip ospf authentication message-digest
ip ospf message-digest-key 1 md5 i<3junos
```

Once entered, you should see the message in the log file stating you have a relationship, and see the IOS router as a neighbor using the Junos `show ospf neighbor` command.

Once you have enabled security on the interfaces in the backbone area, realize the interfaces in Area 0.0.0.1 and Area 0.0.0.2 are not secure, and an attacker could place a router on the switch in either of these areas and flood advertisements for nonexistent subnets.

While you could use the same authentication option, recall from the topology that there are no routers in these areas. OSPF must be enabled in those interfaces for those networks to be advertised, but you can set these interfaces to be what is known as *passive*. This means that the interfaces do not send advertisements out of those interfaces, but do advertise the connected subnets.

Enabling this option within Junos is done like this:

```
edit protocols ospf area 0.0.0.2 interface ge-0/0/1.0
set passive
commit
```

And in IOS the commands would be as follows:

```
router ospf 1
passive-interface fa0/1
```

Now confirm that these networks are now being advertised to their neighbors and appear on the routing table, simply by running the `show route` command:

```
root@J2320> show route

inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.10.0/24      *[Direct/0] 02:42:23
                    > via ge-0/0/0.0
10.10.10.1/32      *[Local/0] 02:42:23
                      Local via ge-0/0/0.0
10.20.20.0/24      *[Direct/0] 00:02:50
                    > via ge-0/0/1.0
10.20.20.1/32      *[Local/0] 00:09:18
                      Local via ge-0/0/1.0
10.30.30.0/24      *[OSPF/10] 00:01:52, metric 2
                    > to 10.10.10.2 via ge-0/0/0.0
```

```
224.0.0.5/32       *[OSPF/10] 02:37:10, metric 1
                      MultiRecv
```

You can also confirm that you are able to ping the interface in Area 0.0.0.1 from a device in Area 0.0.0.2 by specifying the `source <ip address>` after the `ping` command:

```
root@J2320> ping 10.30.30.1 source 10.20.20.1
PING 10.30.30.1 (10.30.30.1): 56 data bytes
64 bytes from 10.30.30.1: icmp_seq=0 ttl=255 time=2.888 ms
64 bytes from 10.30.30.1: icmp_seq=1 ttl=255 time=3.126 ms
64 bytes from 10.30.30.1: icmp_seq=2 ttl=255 time=22.062 ms
64 bytes from 10.30.30.1: icmp_seq=3 ttl=255 time=2.863 ms
```

Recipe 7

Understanding the Host Checker

This recipe walks you through the fundamental principles behind the Host Checker; explaining how different policies can control the level of access a user receives. This is often a confusing area for network administrators, so a simple example policy will be created and then applied.

Problem

How do you ensure that a device logging onto your network has the proper configuration and level of access? Network administrators face a constant dilemma between providing comprehensive access to network resources, while still providing adequate protection. The Juniper remote access and access control platforms are able to differentiate between a healthy corporate device, one which is missing patches, and a totally unknown device. Once the device has been classified, policies can be applied accordingly.

Solution

The Juniper Secure Access, MAG, and Unified Access Controllers provide dynamic access control, whereby polices are created which match users to a set of permissible resources such as Web links, Terminal Servers, or entire subnets. Resource management differs between the Secure Access and Unified Access Controllers, but the host check function works in the same manner on both platforms. Beyond a simple access control list, technologies based upon the IVE platform are also able to factor in the status and type of connections on the connecting endpoint.

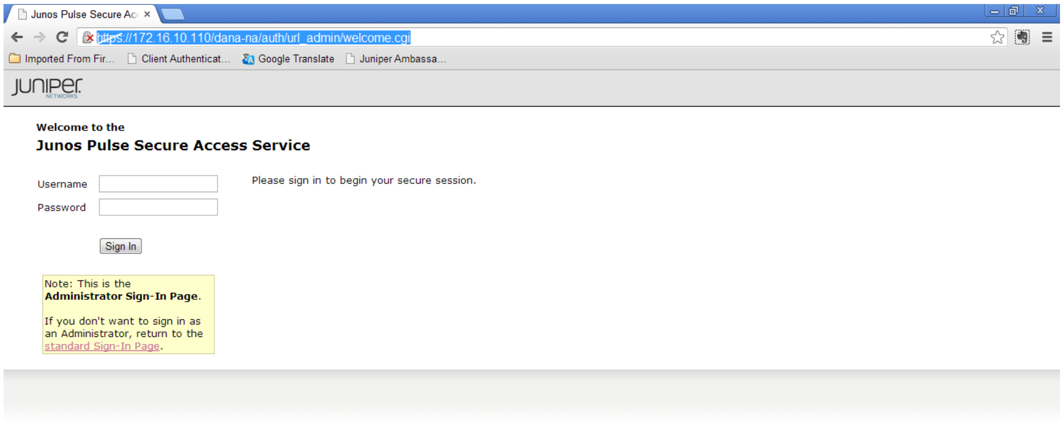
The operating system and web browser capabilities can be inferred at session setup, while the user identity is determined by verifying the user ID against an authentication store such as ActiveDirectory, which is then matched against a directory service. However, to definitively determine the status (and by extension, the health) of the connecting endpoint, an agent is required on the endpoint: the Host Checker.

The Host Checker scans the workstation looking for adherence to an administrator-specified policy. This policy may be something simple, such as the presence of a given application in memory, but it can also be very complex and take into account multiple factors such as Anti-Virus vendor and patch level. The outcomes can be mapped to different levels of access. There are also different approaches to scanning endpoints that are linked closely to the realm and role level policies design. Users can be either be scanned for every eventuality and then broken down at the role level, or scanned only for a subset of policies according to the realm they are authenticating against.

Example: A Simple Policy

Let's create a simple policy to scan for a single file on an endpoint's operating system. The purpose of the check is to determine whether the connecting endpoint is a corporate-built laptop or an otherwise unknown device. This policy can easily be refined to be more specific, but let's start simply so you can better understand how the success or failure of a given test influences the outcome of the policy and resources presented to the user.

1. Access the Administrator Sign-In Page for the Junos Pulse Secure Access Service (SAS) and sign in as an administrator.



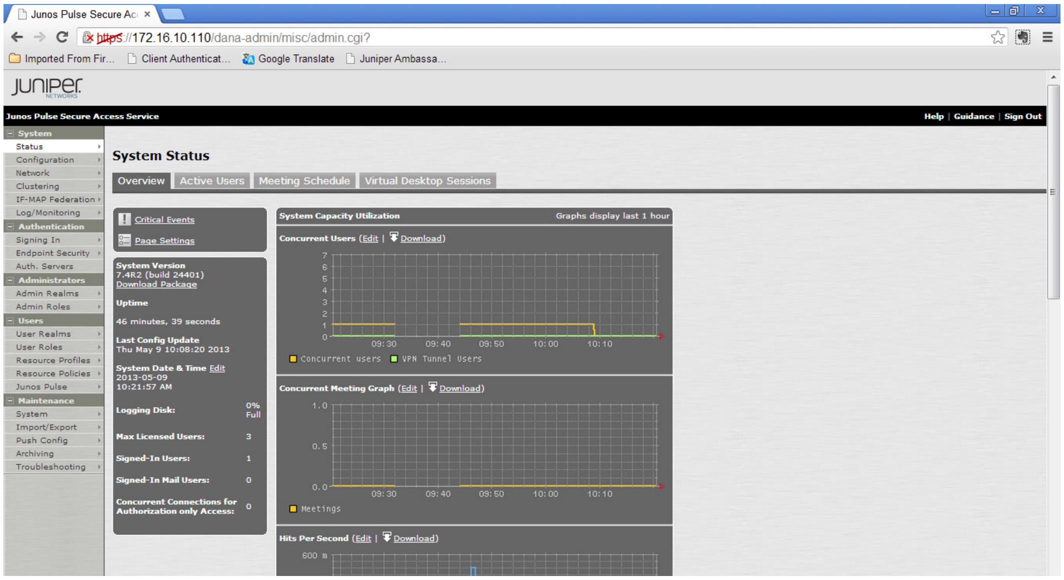
Welcome to the
Junos Pulse Secure Access Service

Username
Password
Please sign in to begin your secure session.
Sign In

Note: This is the **Administrator Sign-In Page**.

If you don't want to sign in as an Administrator, return to the standard Sign-In Page.

2. The System Status page should be displayed.



System Status

System Version
7.4R2 (build 24401)
Download Package

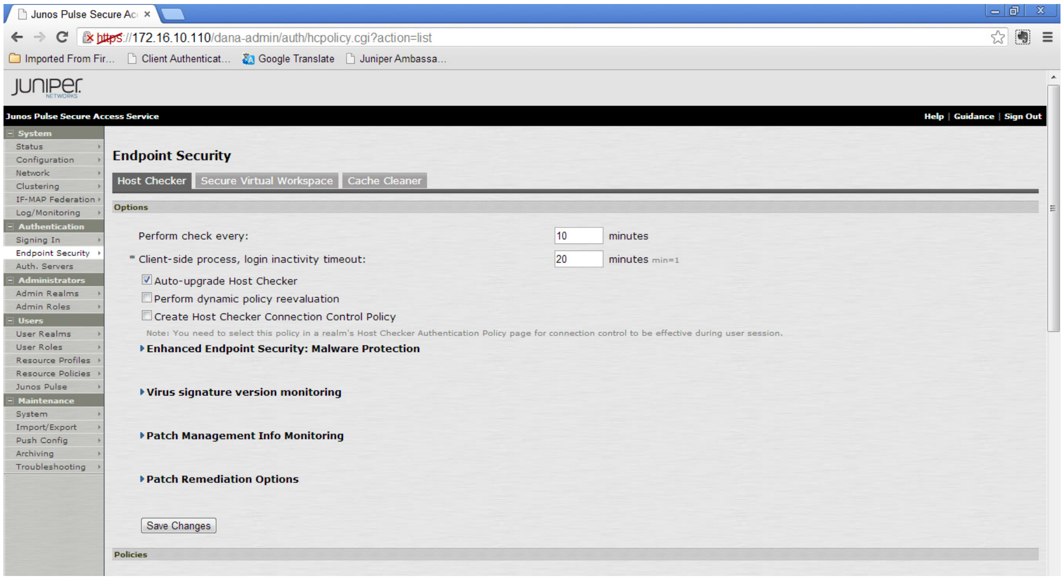
Uptime
46 minutes, 39 seconds

Last Config Update
Thu May 9 10:08:20 2013

System Date & Time Edit
2013-05-09
10:21:57 AM

Logging Disk:	0% Full
Max Licensed Users:	3
Signed-In Users:	1
Signed-In Mail Users:	0
Concurrent Connections for Authorization only Access:	0

3. Navigate to *Endpoint Security* and then the *Host Checker* tab.



Endpoint Security

Options

Perform check every: 10 minutes

* Client-side process, login inactivity timeout: 20 minutes min=1

[✓] Auto-upgrade Host Checker

[ ] Perform dynamic policy reevaluation

[ ] Create Host Checker Connection Control Policy

Note: You need to select this policy in a realm's Host Checker Authentication Policy page for connection control to be effective during user session.

▸ Enhanced Endpoint Security: Malware Protection

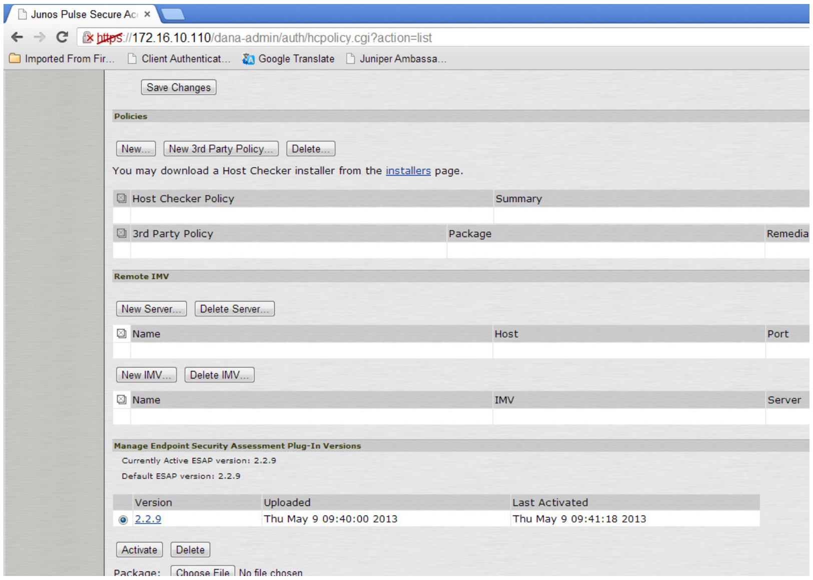
▸ Virus signature version monitoring

▸ Patch Management Info Monitoring

▸ Patch Remediation Options

Policies

4. Scroll down to the *Policies* section and click the *New…* button.


Junos Pulse Secure Ac
~~https~~://172.16.10.110/dana-admin/auth/hcpolicy.cgi?action=list
Imported From Fir… Client Authenticat… Google Translate Juniper Ambassa…
Save Changes
Policies
New… New 3rd Party Policy… Delete…

You may download a Host Checker installer from the installers page.

Host Checker Policy	Summary

3rd Party Policy	Package	Remedia

Remote IMV
New Server… Delete Server…

Name	Host	Port

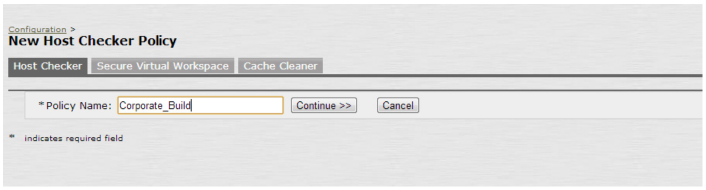
Name	IMV	Server

Manage Endpoint Security Assessment Plug-In Versions
Currently Active ESAP version: 2.2.9
Default ESAP version: 2.2.9

Version	Uploaded	Last Activated
● 2.2.9	Thu May 9 09:40:00 2013	Thu May 9 09:41:18 2013

Activate Delete
Package: Choose File No file chosen

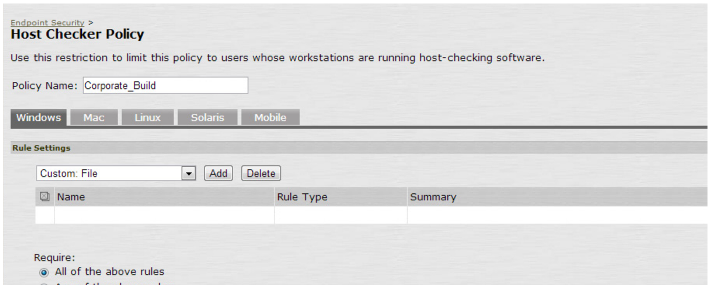
5. The screen will update and prompt for a policy name. Set the name as *Corporate_Build* and select the *Continue* button



*Policy Name: Corporate_Build

* indicates required field

6. This will create a blank policy. Note the tabs which specify different policies for the various supported Host Checker platforms. From the drop-down box in the *Rule Settings Section*, select *Custom:File* and click *Add* to create a new rule.



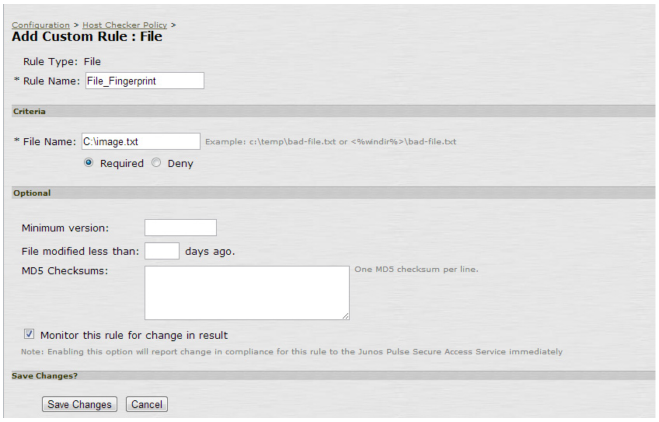
Use this restriction to limit this policy to users whose workstations are running host-checking software.

Policy Name: Corporate_Build

Name	Rule Type	Summary

Require:
● All of the above rules

7. Name the policy rule something memorable, in this case *File_Fingerprint*. This rule is designed to search for a file left over from the disk imaging process commonly used on corporate workstation builds. The contents and location are somewhat irrelevant, as long as they are consistent across all devices that are to be considered trusted. Ensure that the *Required* option is set under the file name. It is also possible to specify additional requirements such as file versioning attributes or an MD5 checksum. Select *Monitor this rule for change in result* to ensure any posture changes on the endpoint security are instantly reported to the Secure Access Service. Click *Save Changes* when complete.



Configuration > Host Checker Policy >
Add Custom Rule : File

Rule Type: File
* Rule Name: File_Fingerprint

Criteria

* File Name: C:\image.txt   Example: c:\temp\bad-file.txt or <%windir%>\bad-file.txt
[✓ Required] Deny

Optional

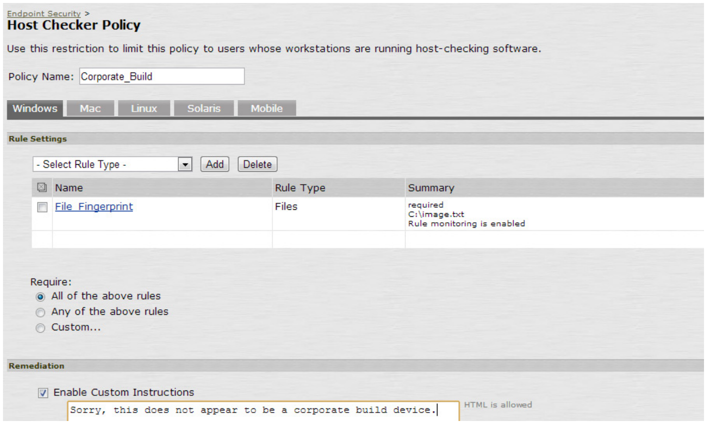
Minimum version:
File modified less than: days ago.
MD5 Checksums: One MD5 checksum per line.

[✓] Monitor this rule for change in result
Note: Enabling this option will report change in compliance for this rule to the Junos Pulse Secure Access Service immediately

Save Changes?

Save Changes | Cancel

8. The Policy window should update with the newly created rule. In the *Remediation* section tick the box next to *Enable Custom Instructions*. This allows the system administrator to provide the enduser with more detailed information or links to a support web site. Enter text to describe the error message to the user and click *Save* at the bottom of the screen.



Endpoint Security >
Host Checker Policy

Use this restriction to limit this policy to users whose workstations are running host-checking software.

Policy Name: Corporate_Build

Windows | Mac | Linux | Solaris | Mobile

Rule Settings

- Select Rule Type - | Add | Delete

Name	Rule Type	Summary
File_Fingerprint	Files	required C:\image.txt Rule monitoring is enabled

Require:
[✓ All of the above rules]
Any of the above rules
Custom...

Remediation

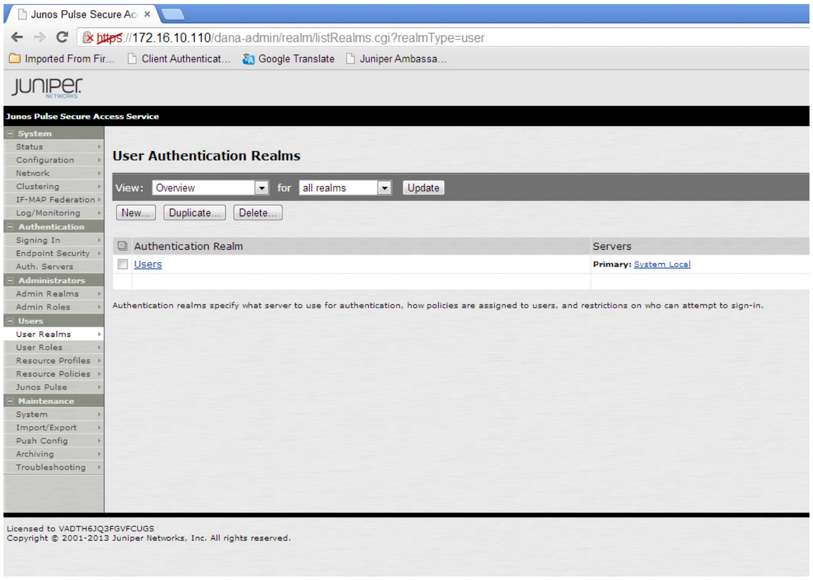
[✓] Enable Custom Instructions
Sorry, this does not appear to be a corporate build device.   HTML is allowed

9. Now that a simple policy has been created, it needs to be bound to a both a Realm and a Role to be assessed, which is covered next.

Working with Realm Level Restrictions

Before a Host Checker Policy can be assessed, it needs to be bound to a Realm. Individual policies can be bound to many different Realms, but in this case the focus will be a single primary Realm.

1. Navigate to the *User Authentication Realms* window from the left-hand tab in the Junos Pulse Secure Access Service Administrator Console. Select the Realm you wish use the Host Checker with.



User Authentication Realms

View: Overview for all realms

Authentication Realm	Servers
Users	Primary: System Local

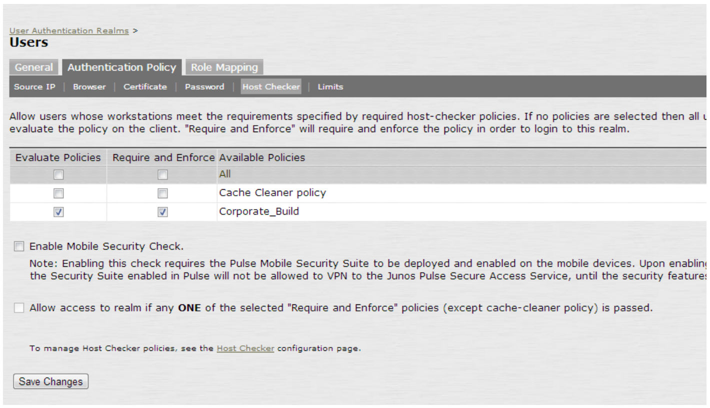
Authentication realms specify what server to use for authentication, how policies are assigned to users, and restrictions on who can attempt to sign-in.

Licensed to VADTH6JQ3FGVFCUGS
Copyright © 2001-2013 Juniper Networks, Inc. All rights reserved.

2. Navigate to the *Authentication Policy* tab and the *Host Check* tab underneath that. All the available host check polices will be listed. There are two options for each policy:

- *Evaluate Policies*: When a user attempts to access this Realm, the host checker will assess the endpoint *Post authentication*.
- *Require and Enforce*: When a user attempts to access a Realm with *Enforce* enabled, the endpoint assessment will be performed *Pre authentication*. If the endpoint fails the assessment, they will be denied access to the realm entirely.

Select both *Evaluate Policies* and *Require and Enforce* on the Corporate_Build policy and click Save Changes to complete the step.


User Authentication Realms >
Users

General | Authentication Policy | Role Mapping

Source IP | Browser | Certificate | Password | Host Checker | Limits

Allow users whose workstations meet the requirements specified by required host-checker policies. If no policies are selected then all u
evaluate the policy on the client. "Require and Enforce" will require and enforce the policy in order to login to this realm.

Evaluate Policies	Require and Enforce	Available Policies
☐	☐	All
☐	☐	Cache Cleaner policy
✓	✓	Corporate_Build

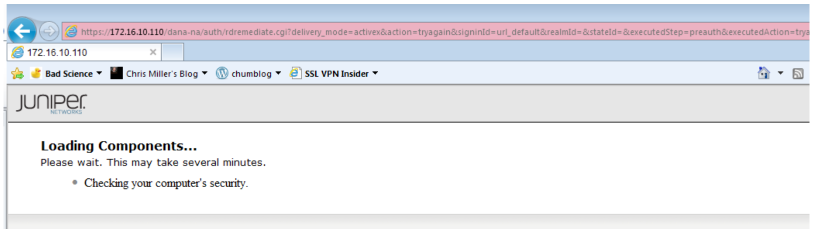
☐ Enable Mobile Security Check.
Note: Enabling this check requires the Pulse Mobile Security Suite to be deployed and enabled on the mobile devices. Upon enablin
the Security Suite enabled in Pulse will not be allowed to VPN to the Junos Pulse Secure Access Service, until the security feature

☐ Allow access to realm if any **ONE** of the selected "Require and Enforce" policies (except cache-cleaner policy) is passed.

To manage Host Checker policies, see the Host Checker configuration page.

Save Changes

3. The next action is to test authentication with the Realm-Level restrictions in place. Navigate to the Default sign-in page of the Junos Pulse Secure Access Service appliance and try to sign in. The browser window will inform the user that it is checking the user's security.


https://172.16.10.110/dana-na/auth/rdremediate.cgi?delivery_mode=activex&action=tryagain&signinId=url_default&realmId=&stateId=&executedStep=preauth&executedAction=trya

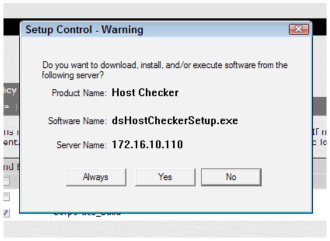
172.16.10.110

Bad Science | Chris Miller's Blog | chumblog | SSL VPN Insider

Loading Components...
Please wait. This may take several minutes.

- Checking your computer's security.

4. If this is the first time the user has attempted to authenticate to the Secure Access Service, and no provision has been made to deploy the Juniper Installer service, it is quite likely they will be prompted as to whether they wish to download and install the host checker package. The user should select *Always* to prevent the prompt from reoccurring on subsequent attempts.


Setup Control - Warning

Do you want to download, install, and/or execute software from the following server?

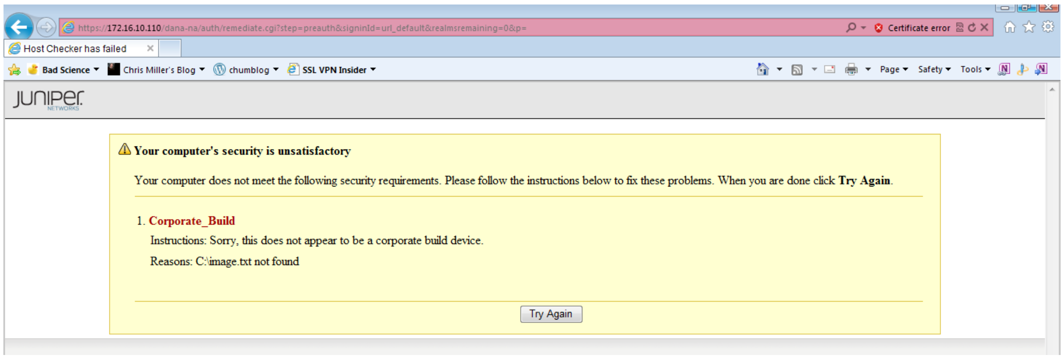
Product Name: Host Checker

Software Name: dsHostCheckerSetup.exe

Server Name: 172.16.10.110

Always | Yes | No

- If the test passes successfully, the user should be returned to the standard authentication prompt, allowing them to authenticate to the Realm.
- If one or more of the tests fails, the user will be prompted with a yellow warning dialogue box. This will list all the tests which have failed, any custom instructions configured, and a detail string describing why the test failed. The only option the user has is *Try Again*; this will force the host checker to rerun the test if the user has successfully addressed the issue.


https://172.16.10.110/dana-na/auth/remediate.cgi?step=preauth&signinId=url_default&realmsremaining=0&p=

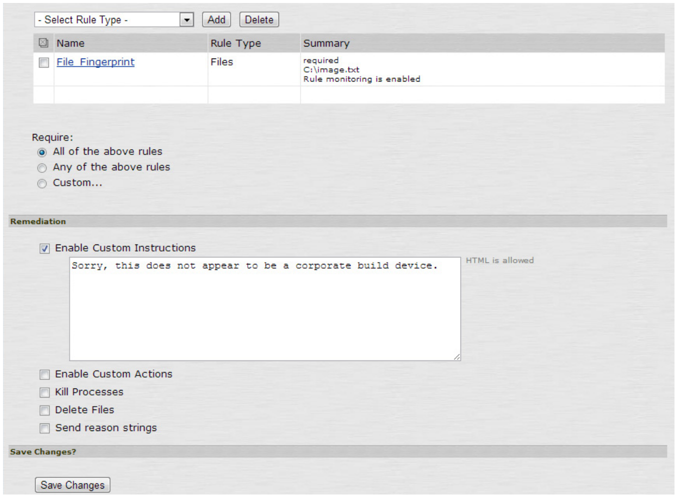
Host Checker has failed

Your computer's security is unsatisfactory

Your computer does not meet the following security requirements. Please follow the instructions below to fix these problems. When you are done click **Try Again**.

1. Corporate_Build
   Instructions: Sorry, this does not appear to be a corporate build device.
   Reasons: C:\image.txt not found

NOTE The *Reason* strings is very helpful when debugging a specific error, but in production it should not enabled as it may provide useful information to someone attempting to break into the environment. To disable the reason strings, return to the *Host Checker* menu in the Junos Pulse Secure Access Service administration console and edit the *Corporate_ Build* Policy. Scroll to the bottom of the Window and remove the tick from the *Send reason strings* box and click *Save Changes*.



	Name	Rule Type	Summary
☐	File_Fingerprint	Files	required C:\image.txt Rule monitoring is enabled

Require:
[✓ All of the above rules]
Any of the above rules
Custom...

Remediation

☑ Enable Custom Instructions

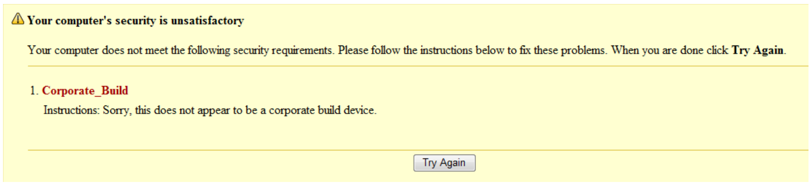
Sorry, this does not appear to be a corporate build device.

HTML is allowed

☐ Enable Custom Actions
☐ Kill Processes
☐ Delete Files
☐ Send reason strings

Save Changes?

This will prevent the Secure Access Service from reporting more information than is necessary.


Your computer's security is unsatisfactory

Your computer does not meet the following security requirements. Please follow the instructions below to fix these problems. When you are done click **Try Again**.

1. **Corporate_Build**
   Instructions: Sorry, this does not appear to be a corporate build device.

As the host checks are run pre-authentication, it is not possible to definitively determine which users are succeeding or failing the tests. However, all attempts are logged and it may be possible to identify the user from the connecting IP address based upon previous access attempts.

Realm-Level Enforcement Failure:

```
2013-05-09 10:43:06 - ive - [172.16.10.11] System()[] - Host Checker policy 'Corporate_
Build' failed on host 172.16.10.11 . Reason: 'C:\image.txt not found'.
```

Realm-Level Enforcement Success, Followed by Authentication:

```
2013-05-09 10:53:28 - ive - [172.16.10.11] System()[] - Host Checker policy 'Corporate_
Build' passed on host 172.16.10.11 .
2013-05-09 11:04:04 - ive - [172.16.10.11] glenk(Users)[] - Primary authentication
successful for glenk/System Local from 172.16.10.11
```

By enforcing the host checks at the realm level, an administrator can be certain that only users with trusted devices are able to authenticate to the Secure Access Service. However, this kind of policy is draconian and does not provide any fine-grained control.

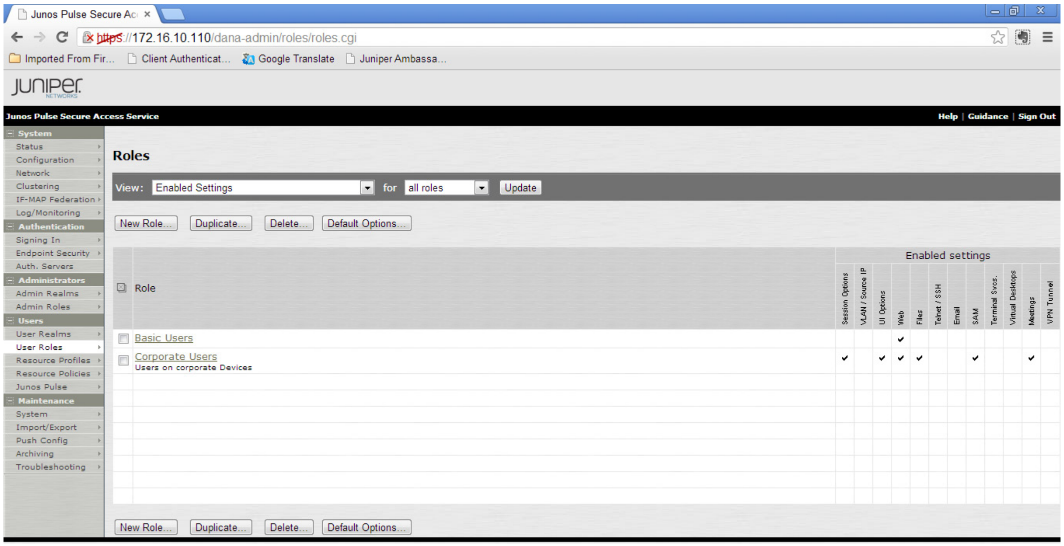
Consider the following. A user wishes to use the VPN and the policy requires that their workstation must be up-to-date with Anti-Virus patterns. The corporate policy dictates that the connecting clients must collect the latest pattern files from a server protected by the Secure Access Device. With Realm-level enforcement in place, there is no opportunity to allow the user to pass authentication and remediate the workstation. To provide a workaround, role level restrictions can be used to provide limited access to allow the endpoint to be updated without exposing the organization to any additional security risks.

Working with Role Level Restrictions

Up to now, a very broad approach was used to grant or deny access to the Secure Access Service. With Role level restrictions a much more sophisticated and flexible policy can be created.

1. Within the Junos Pulse Secure Access Service administrator interface, navigate to the *User Roles* Menu. This displays all currently created roles on the appliance. In the example here, the *Basic Users*

role only has a single Web bookmark defined, while the *Corporate Users* role has numerous enabled resources. Click on the *Corporate Users* role.



Junos Pulse Secure Access Service

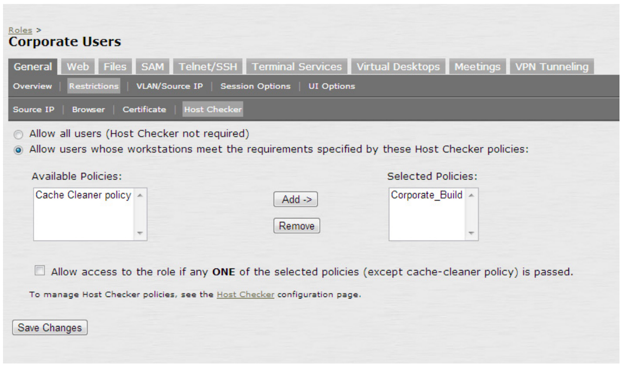
Roles

View: Enabled Settings for all roles Update

New Role... Duplicate... Delete... Default Options...

Role	Session Options	VLAN / Source IP	UI Options	Web	Files	Telnet / SSH	Email	SAM	Terminal Svcs.	Virtual Desktops	Meetings	VPN Tunnel
Basic Users				✓								
Corporate Users Users on corporate Devices	✓		✓	✓	✓			✓			✓	

2. Under the *Corporate Users* role, navigate to the *General* tab, *Restrictions* and then the *Host Checker* tab. Change the radio button to *Allow users whose workstations meet…* and add the *Corporate_Build* policy to the *Selected Policies* list. Click the *Save Changes* button to commit the configuration.



Roles >
Corporate Users

General | Web | Files | SAM | Telnet/SSH | Terminal Services | Virtual Desktops | Meetings | VPN Tunneling

Overview | Restrictions | VLAN/Source IP | Session Options | UI Options

Source IP | Browser | Certificate | Host Checker

○ Allow all users (Host Checker not required)
● Allow users whose workstations meet the requirements specified by these Host Checker policies:

Available Policies: Cache Cleaner policy

Add -> Remove

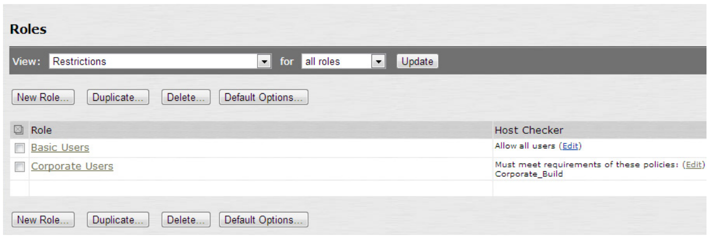
Selected Policies: Corporate_Build

☐ Allow access to the role if any **ONE** of the selected policies (except cache-cleaner policy) is passed.

To manage Host Checker policies, see the Host Checker configuration page.

Save Changes

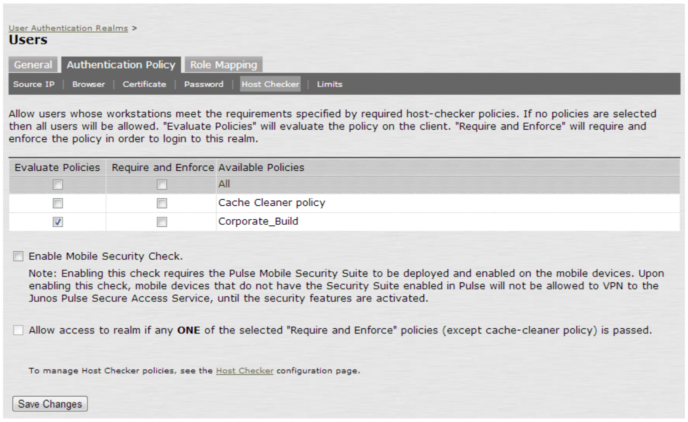
3. Return to the *User Roles* page. In the *View:* drop down box select the *Restrictions* option from the list and select the *Update* button to change the view. This will show all the roles and any restrictions placed upon them. In this case, users granted access to the *Corporate Users* role must first satisfy the requirements of the *Corporate_Build* Host Checker policy.


Roles
View: Restrictions for all roles Update
New Role... Duplicate... Delete... Default Options...

Role	Host Checker
Basic Users	Allow all users (Edit)
Corporate Users	Must meet requirements of these policies: (Edit) Corporate_Build

New Role... Duplicate... Delete... Default Options...

4. Navigate to the target authentication realm and select the *Authentication Policy* tab and then the *Host Checker* tab. Uncheck the *Require and Enforce* box on the *Corporate_Build* policy before saving the changes.


User Authentication Realms >
Users
General | Authentication Policy | Role Mapping
Source IP | Browser | Certificate | Password | Host Checker | Limits

Allow users whose workstations meet the requirements specified by required host-checker policies. If no policies are selected then all users will be allowed. "Evaluate Policies" will evaluate the policy on the client. "Require and Enforce" will require and enforce the policy in order to login to this realm.

Evaluate Policies	Require and Enforce	Available Policies
☐	☐	All
☐	☐	Cache Cleaner policy
☑	☐	Corporate_Build

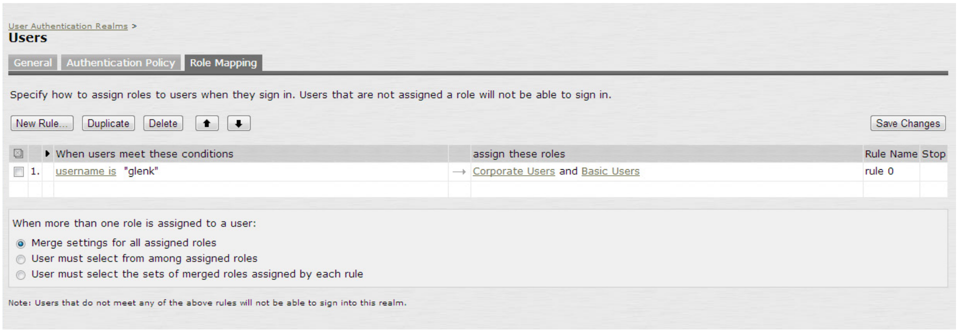
☐ Enable Mobile Security Check.
Note: Enabling this check requires the Pulse Mobile Security Suite to be deployed and enabled on the mobile devices. Upon enabling this check, mobile devices that do not have the Security Suite enabled in Pulse will not be allowed to VPN to the Junos Pulse Secure Access Service, until the security features are activated.

☐ Allow access to realm if any **ONE** of the selected "Require and Enforce" policies (except cache-cleaner policy) is passed.

To manage Host Checker policies, see the Host Checker configuration page.

NOTE With this configuration, when a user attempts to authenticate against the realm, they are subjected to post-authentication host checks. However, the resources they will be able to access are dependent upon the Role level restrictions in place.

5. The next stage is grant access to the two roles, one with, and one without, restrictions. Navigate to the *Role Mapping* tab of the target *User Authentication Realm*. Create a new rule that maps the user or groups to both roles. Once authentication has been satisfied then the Secure Access Service will map the user to all roles other than those with unmet restrictions.


User Authentication Realms >
Users

General | Authentication Policy | Role Mapping

Specify how to assign roles to users when they sign in. Users that are not assigned a role will not be able to sign in.

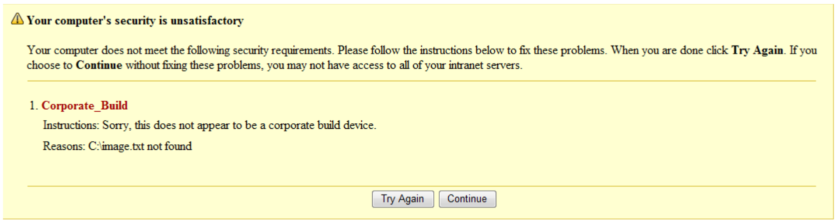
New Rule... | Duplicate | Delete | Save Changes

		When users meet these conditions		assign these roles	Rule Name	Stop
☐	1.	username is "glenk"	→	Corporate Users and Basic Users	rule 0	

When more than one role is assigned to a user:
- [✓ Merge settings for all assigned roles]
- User must select from among assigned roles
- User must select the sets of merged roles assigned by each rule

Note: Users that do not meet any of the above rules will not be able to sign into this realm.

6. Remove the c:\image.txt file from the test workstation and attempt to authenticate. The user still receives the same *Your computer's security is unsatisfactory* warning message, but this time is able to continue to the next stage.

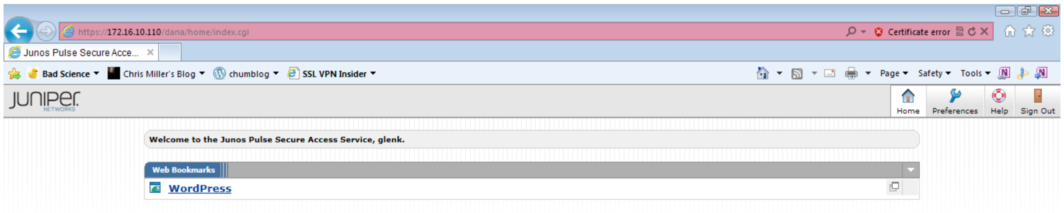

Your computer's security is unsatisfactory

Your computer does not meet the following security requirements. Please follow the instructions below to fix these problems. When you are done click **Try Again**. If you choose to **Continue** without fixing these problems, you may not have access to all of your intranet servers.

1. **Corporate_Build**
   Instructions: Sorry, this does not appear to be a corporate build device.
   Reasons: C:\image.txt not found

Try Again | Continue

7. Once the user presses the continue button, they are presented with the resources attached to the Roles they have been granted access to. In this case the user was only permitted access to resources in the Basic Users Role.


https://172.16.10.110/dana/home/index.cgi

Certificate error

Junos Pulse Secure Acce...

Bad Science | Chris Miller's Blog | chumblog | SSL VPN Insider | Page | Safety | Tools

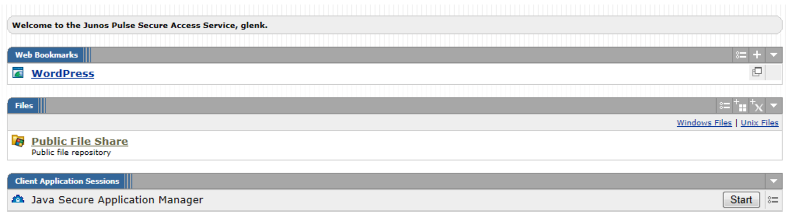
Home | Preferences | Help | Sign Out

Welcome to the Junos Pulse Secure Access Service, glenk.

Web Bookmarks

WordPress

8. Restore the C:\image.txt file and sign-in to the Secure Access service again. Once the host checks and authentication have been satisfied the user should have visibility of all the assigned roles and resources.


Welcome to the Junos Pulse Secure Access Service, glenk.

Web Bookmarks
WordPress

Files
Windows Files | Unix Files
Public File Share
Public file repository

Client Application Sessions
Java Secure Application Manager Start

Running the host checks post authentication creates a more flexible policy, allowing users some level of access even if they fail one or more host checks. However, the downside is that by accepting the user credentials before the health of the end-point has been assessed they could be captured by malicious software before forwarding to the Secure Access Service – theoretically this could be avoided by running an advanced anti-malware host check pre-authentication.

By running the host checks post-authentication, it is also possible to get a clear view of which users are attempting to authenticate and why they are failing the test. The following log entries clearly show the successes and failures.

Role-Level enforcement failure:

```
2013-05-09 11:20:04 - ive - [172.16.10.11] glenk(Users)[] - Primary authentication
successful for glenk/System Local from 172.16.10.11
2013-05-09 11:20:08 - ive - [172.16.10.11] glenk(Users)[] - Host Checker policy
'Corporate_Build' failed on host 172.16.10.11 . Reason: 'C:\image.txt not found'.
2013-05-09 11:25:21 - ive - [172.16.10.11] glenk(Users)[Basic Users] - Login succeeded
for glenk/Users (session:00000000) from 172.16.10.11.
```

These three events clearly show the user correctly authentication against into the *Users* Realm, failing the *Corporate Build* host check, and then being granted access to the unrestricted *Basic Users* role.

Role-Level enforcement successes:

```
2013-05-09 16:48:24 - ive - [172.16.10.11] glenk(Users)[] - Primary authentication
successful for glenk/System Local from 172.16.10.11
2013-05-09 16:48:28 - ive - [172.16.10.11] glenk(Users)[] - Host Checker policy
'Corporate_Build' passed on host 172.16.10.11 for user 'glenk'.
2013-05-09 16:48:30 - ive - [172.16.10.11] glenk(Users)[Corporate Users, Basic Users]
- Login succeeded for glenk/Users (session:00000000) from 172.16.10.11.
```

These log entries show the successful authentication, the successful host check, and the subsequent access granted to both the restricted *Corporate Users* and *Basic Users* role.

Recipe 8

IP Tracking for Dual ISPs on a SRX

Problem

As Internet links become less expensive you'll see more and more sites with dual ISP connections. Often this is implemented with a high quality main link and a (cheaper) backup line. When DSL or similar connections are used, failure can occur at the WAN side of a router or DSL modem. A simple backup route in the routing table doesn't help in this case because the Ethernet side of the device stays up when the WAN side fails, so our SRX keeps the main route active. A nice way of solving this problem is "tracking" an upstream device such as the ISP's router.

Solution

On Junos devices you can use the RPM (Realtime Performance Monitoring: don't get confused RedHat users) implementation to do the monitoring for you. Now you can set up full RPM with latency monitoring, or even URL tracking, but you can also set up a "simple" tracking with a ping packet. For ISP tracking and failover this is often enough. Let's build an example.

In the example below RPM and IP-monitoring are shown to react to a loss over the primary ge-0/0/1 on srxA-1. The failure occurs not because of a direct connection but because of the loss of srxA-2 's ge-0/0/3 connection to the Internet.

Junos version tested on: 12.1r3

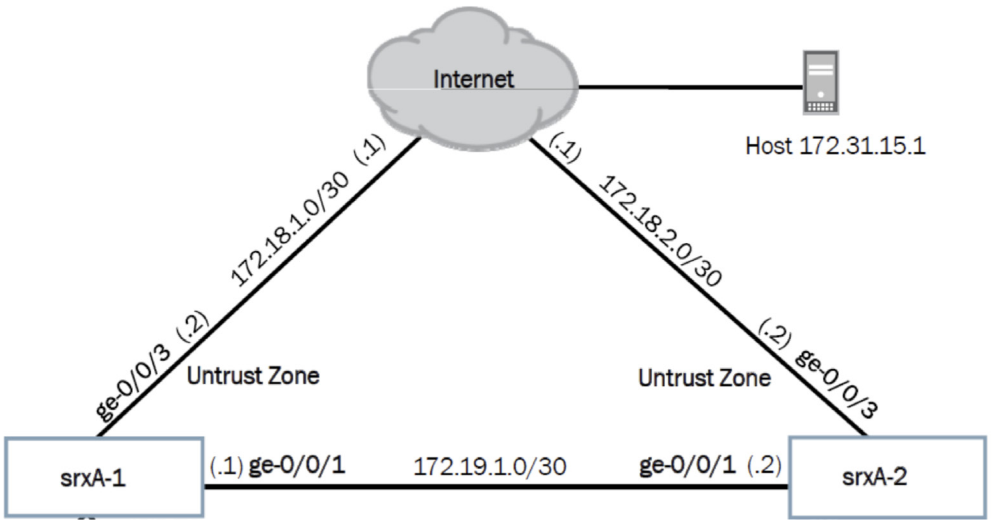


Figure 8.1 Sample Network for this Recipe

During normal operation the route to the Internet should be over next-hop 172.19.1.0. But when the link between srxA-2 and the Internet goes down, then next-hop should change to 172.18.1.1, providing a backup-path this way. On srxA-1, in the configuration in the routing-options static hierarchy, the following routes are configured to support this failover:

```
route 0.0.0.0/0 {
    next-hop 172.19.1.2;
    qualified-next-hop 172.18.1.1 {
        metric 100;
    }
}
```

This will provide a backup route that is activated when ge-0/0/1 on srxA-1 goes down. For the indirect failure (behind the first hop), RPM will be configured later. Accept an active default route pointing to 172.19.1.2 now, and a less preferred route to 172.18.1.1, because of the higher metric value. Let's check this in the routing table with the operational `show route 0/0 exact` command:

```
0.0.0.0/0          *[Static/5] 01:05:24
                    > to 172.19.1.2 via ge-0/0/1.0
                    [Static/5] 00:03:21, metric 100
                    > to 172.18.1.1 via ge-0/0/3.0
```

Things are looking good! The asterisk makes clear 172.19.1.2 is the active next-hop.While the qualified next-hop to 172.18.1.1 via ge-0/0/3.0 isn't strictly necessary because the tracking solution installs a route when the probe fails. It's best practice to see the backup route during normal operations so you can react more quickly to a local link (Ethernet) failure.

Let's configure the RPM probe this way using a ICMP (ping) probe in the services rpm hierarchy:

```
root@srxA-1# show services rpm
probe Probe-Server {
    test testroute {
        probe-type icmp-ping;
        target address 172.18.2.1;
        probe-count 3;
        probe-interval 1;
        test-interval 1;
        thresholds {
            successive-loss 3;
            total-loss 3;
        }
        destination-interface ge-0/0/1.0;
        next-hop 172.19.1.2;
    }
}
```

So you can see the config is probing three times (`probe-count 3`) every second (`probe-interval` 1) with a ping (`probe-type icmp-ping`) to up-stream device 172.18.2.1 . The probe starts again after a second (`test-interval 1`). The configuration gives a probe failure after three missed pings in a row.

Before configuring a reaction to the probe, it might be a good idea to check on how the probe is doing. With a normally functioning link you should see this in our recipe's topology:

```
root@srxA-1> show services rpm probe-results
    Owner: Probe-Server, Test: testroute
    Target address: 172.18.2.1, Probe type: icmp-ping
    Destination interface name: ge-0/0/1.0
    Test size: 3 probes
    Probe results:
      Response received, Mon Oct 22 11:39:45 2012, No hardware timestamps
      Rtt: 1984 usec
    Results over current test:
      Probes sent: 1, Probes received: 1, Loss percentage: 0
      Measurement: Round trip time
        Samples: 1, Minimum: 1984 usec, Maximum: 1984 usec, Average: 1984 usec,
        Peak to peak: 0 usec, Stddev: 0 usec, Sum: 1984 usec
    Results over last test:
      Probes sent: 3, Probes received: 3, Loss percentage: 0
      Test completed on Mon Oct 22 11:39:44 2012
      Measurement: Round trip time
```

```
        Samples: 3, Minimum: 1959 usec, Maximum: 1985 usec, Average: 1971 usec,
        Peak to peak: 26 usec, Stddev: 11 usec, Sum: 5913 usec
    Results over all tests:
      Probes sent: 346, Probes received: 49, Loss percentage: 85
      Measurement: Round trip time
        Samples: 49, Minimum: 1695 usec, Maximum: 4085 usec, Average: 2025 usec,
        Peak to peak: 2390 usec, Stddev: 336 usec, Sum: 99228 usec

root@srxA-1> **show services rpm history-results**
    Owner, Test                 Probe received              Round trip time
    Probe-Server, testroute     Mon Oct 22 11:39:44 2012             1985 usec
    Probe-Server, testroute     Mon Oct 22 11:39:45 2012             1984 usec
    Probe-Server, testroute     Mon Oct 22 11:39:46 2012             2588 usec
    Probe-Server, testroute     Mon Oct 22 11:39:47 2012             2074 usec
…..
```

Now let's create a failure by removing the cable from interface ge-0/0/3 from srxA-2. Now check the results:

```
root@srxA-1> **show services rpm probe-results**
    Owner: Probe-Server, Test: testroute
    Target address: 172.18.2.1, Probe type: icmp-ping
    Destination interface name: ge-0/0/1.0
    Test size: 3 probes
    Probe results:
      Request timed out, Mon Oct 22 11:41:11 2012
    Results over current test:
      Probes sent: 1, Probes received: 0, Loss percentage: 100
    Results over last test:
      Probes sent: 3, Probes received: 0, Loss percentage: 100
    Results over all tests:
      Probes sent: 430, Probes received: 125, Loss percentage: 70
      Measurement: Round trip time
        Samples: 125, Minimum: 1621 usec, Maximum: 10267 usec,
        Average: 2101 usec, Peak to peak: 8646 usec, Stddev: 934 usec,
        Sum: 262634 usec
root@srxA-1> show services rpm history-results
    Owner, Test                 Probe received              Round trip time
    Probe-Server, testroute     Mon Oct 22 11:40:32 2012             1946 usec
    Probe-Server, testroute     Mon Oct 22 11:41:03 2012  Request timed out
    Probe-Server, testroute     Mon Oct 22 11:41:05 2012  Request timed out
         ……………..
```

Okay, the probe works, so now let's configure how to react to this failure using `services ip-monitoring`:

```
root@srxA-1# **show services ip-monitoring**
policy Server-Tracking {
    match {
        rpm-probe Probe-Server;
    }
    then {
        preferred-route {
            route 0.0.0.0/0 {
                next-hop 172.18.1.1;
            }
        }
    }
}
```

A failure in the Internet link (pulled cable from srxA-2's ge-0/0/3 interface) issues this after about five seconds:

```
root@srxA-1> show route
…….
0.0.0.0/0          *[Static/1] 00:00:15, metric2 0
                    > to 172.18.1.1 via ge-0/0/3.0
                    [Static/5] 01:25:36
                    > to 172.19.1.2 via ge-0/0/1.0
                    [Static/5] 00:23:33, metric 100
                    > to 172.18.1.1 via ge-0/0/3.0
```

Now, when the probe detects a failure, the ip-monitor installs a route with a preference of 1. This high preference makes sure our backup is chosen over every static route (static routes have a default preference of 5).

The output makes clear our goal of changing a route because of indirect failure is reached by the config we created.

MORE? For more information about the technology used here please refer to the links below: http://kb.juniper.net/InfoCenter/index?page=content&id=KB22052&actp=search&viewlocale=en_US&searchid=1350650857447. For manual page rpm: http://www.juniper.net/techpubs/en_US/junos12.1/topics/task/configuration/rpm-probes-configuring.html.

And for those who don't know how to use `load merge relative terminal`, here is the used configuration:

```
set services rpm probe Probe-Server test testroute probe-type icmp-ping
set services rpm probe Probe-Server test testroute target address 172.18.2.1
set services rpm probe Probe-Server test testroute probe-count 3
set services rpm probe Probe-Server test testroute probe-interval 1
set services rpm probe Probe-Server test testroute test-interval 1
set services rpm probe Probe-Server test testroute thresholds successive-loss 3
set services rpm probe Probe-Server test testroute thresholds total-loss 3
set services rpm probe Probe-Server test testroute destination-interface ge-0/0/1.0
set services rpm probe Probe-Server test testroute next-hop 172.19.1.2
set services ip-monitoring policy Server-Tracking match rpm-probe Probe-Server
set services ip-monitoring policy Server-Tracking then preferred-
route route 0.0.0.0/0 next-hop 172.18.1.1
```

Recipe 9

Configuring Multicast with BSR

The Bootstrap Router (BSR) protocol is used in a PIM Sparse-Mode (PIM-SM) environment to signal the location of the rendezvous-points (RPs) to all routers participating in the multicast network. This recipe explains how multicast can be configured in an enterprise using BSR, by configuring it with BSR for both IPv4 and IPv6 networks.

Problem

PIM sparse mode is desired in a multicast network, which requires a rendezvous-point (RP). The RP can be created statically or by using a dynamic protocol such as Auto-RP or BSR. Since Auto-RP requires sparse-dense mode, BSR has been chosen as the protocol to carry RP information to the routers in the network

The topology for this section consists of four routers connected in a ring. OSPF area 0 is running between the routers, with the fe-0/0/2 interfaces included in OSPF (passive). There is a single multicast sender and a single multicast receiver sending a multicast stream for IPv4 group 239.1.1.1 and IPv6 group FF1E::1.

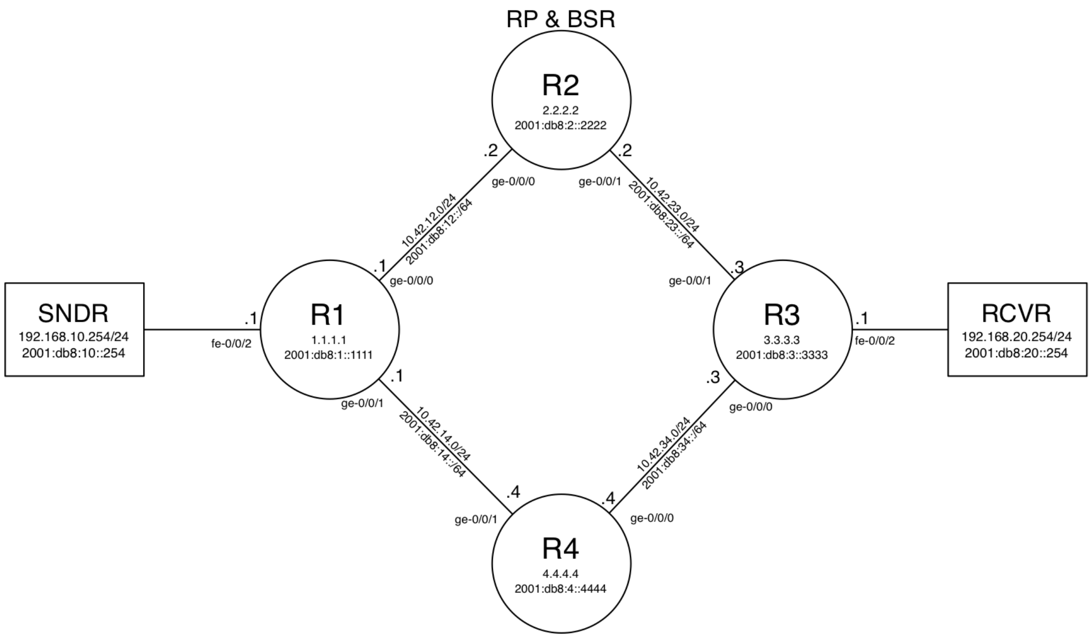


Figure 9.1 Multicast with BSR Topology

Solution

Configuration of BSR in a network is a fairly simple process. A router acting as the RP should be chosen, as well as a router performing the BSR duties. The same router can perform both roles (as in this example), or separate routers can perform them.

To configure multicast with BSR, the following steps should be taken:

- Configure PIM Sparse-Mode
- Configure the Rendezvous Point (RP)
- Configure the Bootstrap Router (BSR)
- Enable the multicast sender and receiver
- Verify multicast with BSR is functioning

To configure PIM Sparse-Mode:

1. Log in to the router, and enter configuration mode (it's done the same way for each of the routers, so only R1 is shown for steps 1-3).

```
root@R1% cli
root@R1> configure
root@R1#
```

2. Configure PIM Sparse-Mode (PIM-SM), which is done under the `[edit protocols pim]` hierarchy.

```
[edit]
root@R1# edit protocols pim

[edit protocols pim]
root@R1# set interface all mode sparse
```

3. Verify the PIM configuration on all four routers. Each configuration should be identical, and R1's configuration is shown here.

```
[edit protocols pim]
root@R1# show
interface all {
    mode sparse;
}
```

4. Next, configure R2 to be the Rendezvous Point (RP). This configuration is found under the `[edit protocols pim rp local]` hierarchy. Use the family inet and family inet6 loopback addresses.

```
[edit protocols pim]
root@R2# set rp local family inet address 2.2.2.2

[edit protocols pim]
root@R2# set rp local family inet6 address 2001:db8:2::2222
```

5. Next, add the BSR configuration to R2. By default, the bootstrap priority for all routers is 0. By configuring the priority of 100 for both address families, R2 is guaranteed to become the BSR for the topology. This configuration is done under the `[edit protocols pim rp bootstrap]` hierarchy.

```
[edit protocols pim]
root@R2# set rp bootstrap family inet priority 100

[edit protocols pim]
root@R2# set rp bootstrap family inet6 priority 100
```

6. Verify the PIM configuration on R2, ensuring the RP and BSR configuration is correct.

```
[edit protocols pim]
root@R2# show
rp {
    bootstrap {
        family inet {
            priority 100;
        }
        family inet6 {
            priority 100;
        }
    }
    local {
        family inet {
            address 2.2.2.2;
        }
        family inet6 {
            address 2001:db8:2::2222;
        }
    }
}
interface all {
    mode sparse;
}
```

7. Jump to the top of the hierarchy and commit the configuration. This step obviously needs to be done on all four routers, but only R1 is shown here.

```
[edit protocols pim]
root@R1# top

[edit]
root@R1# commit and-quit
commit complete
Exiting configuration mode

root@R1>
```

To enable the multicast sender and receiver:

VideoLAN makes a fantastic product called VLC, which can easily be configured to send and receive multicast streams.

MORE? Downloading and installing VLC is beyond the scope of this book, but more information can be found at http://www.videolan.org/vlc/index.html.

1. First, enable an IPv4 multicast stream on SNDR. The command to send a video multicast stream to 239.1.1.1 on UDP/5004 is:

```
vlc video.avi --sout '#rtp{dst=239.1.1.1,port=5004,mux=ts,ttl=10}'
```

2. Next, enable an IPv6 multicast stream on SNDR. The command to send a video multicast stream to FF1E::1 on UDP/5004 is:

```
vlc video.avi --sout '#rtp{dst=ff1E::1,port=5004,mux=ts,ttl=10}'
```

3. Now configure RCVR to receive the IPv4 multicast stream. The command to do this is:

```
vlc rtp://@239.1.1.1:5004
```

4. Finally, configure RCVR to receive the IPv6 multicast stream. The command to do this is:

```
vlc rtp://@[ff1e::1]:5004
```

To verify that multicast with BSR is functioning correctly:

The multicast receivers will provide immediate feedback about whether the streams are working correctly, because the videos will now be playing. Don't get too distracted by watching your favorite episode of Doctor Who playing in ASCII characters, because we still have some more verification to do!

First, verify the output of 'show pim bootstrap' on each of the routers. R1, R3, and R4 should all show a bootstrap priority of 0, making them *InEligible*. R2 should show a priority of 100 and be *Elected*. R1 is the only *InEligible* router shown here.

```
root@R1> show pim bootstrap
Instance: PIM.master

BSR                     Pri Local address           Pri State      Timeout
2.2.2.2                 100 4.4.4.4                   0 InEligible     126
2001:db8::2222          100 2001:db8::4444            0 InEligible      95

root@R2> show pim bootstrap
Instance: PIM.master

BSR                     Pri Local address           Pri State      Timeout
2.2.2.2                 100 2.2.2.2                 100 Elected          6
2001:db8::2222          100 2001:db8::2222          100 Elected         34
```

2. Next, verify that the RPs are discovered by all other routers in the network. On R1, R3, and R4 the RPs will be learned via bootstrap. On R2, the RPs will be learned both via bootstrap and via static.

```
root@R1> show pim rps
Instance: PIM.master
Address family INET
RP address               Type        Holdtime Timeout Groups Group prefixes
2.2.2.2                  bootstrap        150     136      0 224.0.0.0/4

Address family INET6
RP address               Type        Holdtime Timeout Groups Group prefixes
2001:db8:2::2222         bootstrap        150     104      0 ff00::/8
```

```
root@R2> show pim rps
Instance: PIM.master
Address family INET
RP address               Type        Holdtime Timeout Groups Group prefixes
2.2.2.2                  bootstrap        150    None      1 224.0.0.0/4
2.2.2.2                  static             0    None      1 224.0.0.0/4

Address family INET6
RP address               Type        Holdtime Timeout Groups Group prefixes
2001:db8:2::2222         bootstrap        150    None      1 ff00::/8
2001:db8:2::2222         static             0    None      1 ff00::/8
```

3. Verify on R4 that both the (*,G) RPT and the (S,G) SPT are joined. The upstream interface should match the IGP next-hop. In this case, it is ge-0/0/1 towards R2.

```
root@R4> show pim join
Instance: PIM.master Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 239.1.1.1
    Source: *
    RP: 2.2.2.2
    Flags: sparse,rptree,wildcard
    Upstream interface: ge-0/0/1.0

Group: 239.1.1.1
    Source: 192.168.10.254
    Flags: sparse,spt
    Upstream interface: ge-0/0/1.0

Instance: PIM.master Family: INET6
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: ff1e::1
    Source: *
    RP: 2001:db8:2::2222
    Flags: sparse,rptree,wildcard
    Upstream interface: ge-0/0/1.0

Group: ff1e::1
    Source: 2001:db8:10::254
    Flags: sparse,spt
    Upstream interface: ge-0/0/1.0
```

4. Verify the PIM joins on R2. Again, there should be both an RPT and an SPT join. Since this router is the RP, the upstream interface for the (*,G) RPT is 'Local'. Also, the upstream interface for the (S,G) RPT once again follows the IGP next-hop, towards R1.

```
root@R2> show pim join
Instance: PIM.master Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard
```

```
Group: 239.1.1.1
    Source: *
    RP: 2.2.2.2
    Flags: sparse,rptree,wildcard
    Upstream interface: Local

Group: 239.1.1.1
    Source: 192.168.10.254
    Flags: sparse,spt
    Upstream interface: ge-0/0/0.0

Instance: PIM.master Family: INET6
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: ff1e::1
    Source: *
    RP: 2001:db8:2::2222
    Flags: sparse,rptree,wildcard
    Upstream interface: Local

Group: ff1e::1
    Source: 2001:db8:10::254
    Flags: sparse,spt
    Upstream interface: ge-0/0/0.0
```

5. Now verify the PIM joins on R1. Notice that on this router there are no (*,G) RPT joins. That is because this router is directly connected to the source (SNDR) and therefore has no reason to join the RPT.

```
root@R1> show pim join
Instance: PIM.master Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 239.1.1.1
    Source: 192.168.10.254
    Flags: sparse,spt
    Upstream interface: fe-0/0/2.0

Instance: PIM.master Family: INET6
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: ff1e::1
    Source: 2001:db8:10::254
    Flags: sparse,spt
    Upstream interface: fe-0/0/2.0
```

6. Finally, verify that multicast traffic is exiting R4 towards the RCVR. This can be done simply by verifying interface counters:

```
root@R4> show interfaces fe-0/0/2.0 extensive | match Output | match pps
     Output packets:            139784379                  271 pps
```

Discussion

When deciding which protocol to run in order to distribute information about the RP in a PIM Sparse-Mode network, a number of factors should be considered.

- Static RP configurations are simple, but require significant management overhead. If a change in RP address is required, an administrator must change the RP address on every router.
- It must be decided whether there should be any dense-mode groups configured on the network. Auto-RP runs in a sparse-mode environment, but requires a pair of multicast groups to operate in dense mode in order to distribute information about the location of the RP. This may not be desirable.
- BSR provides a standards-based protocol (RFC 5059) that is easy to configure, has low maintenance, and scales well.

More Information

Configuring PIM Bootstrap Router

http://www.juniper.net/techpubs/en_US/junos11.4/topics/topic-map/mcast-pim-bootstrap-router.html

RFC 5059 - Bootstrap Router (BSR) Mechanism for Protocol Independent Multicast (PIM):

http://tools.ietf.org/html/rfc5059

Recipe 10

Optimizing VPN Tunneling Resource Policies

Problem

How do you provide on-network access for remote users while ensuring that they get the best user experience? Can Network Connect and Junos Pulse provide fine-grain control over resources access, how traffic is handled, and how connection parameters are negotiated?

Discussion

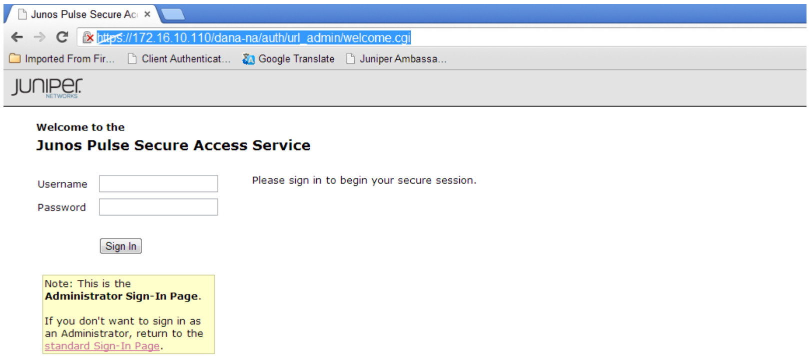
The Juniper Secure Access and MAG (the IVE platform) have two distinct Layer 3 VPN clients: Network Connect and the Junos Pulse Client. Network Connect is typically used in portal-integrated deployments while Pulse is used as a standalone desktop-integrated application.

This recipe walks you through the fundamental principles behind the resource policies used by Network Connect and the Junos Pulse Client, explaining how the policies interact to control the connectivity a user receives. There are several factors to consider when setting up VPN Tunneling policies, so let's create a simple policy to allow a user access to specific subnets. There are four different policies used by Network Connect and Pulse, two of which are mandatory.

Creating an Access Policy

The access policy determines which resources a given role should be granted access to. As per other resources, access rights are cumulative until a specific deny is reached or there are no more policy matches. You may create multiple policies per role and it's quite possible you may have many policies for different levels of access. For example, third-party contractors may only have access to specific hosts, normal users may be granted access to whole networks but with specific exclusions, and network administrators may be granted access to all accessible resources.

1. To begin, access the Administrator Sign In Page for the Junos Pulse Secure Access Service (SAS) and sign in as an administrator.



Welcome to the
Junos Pulse Secure Access Service

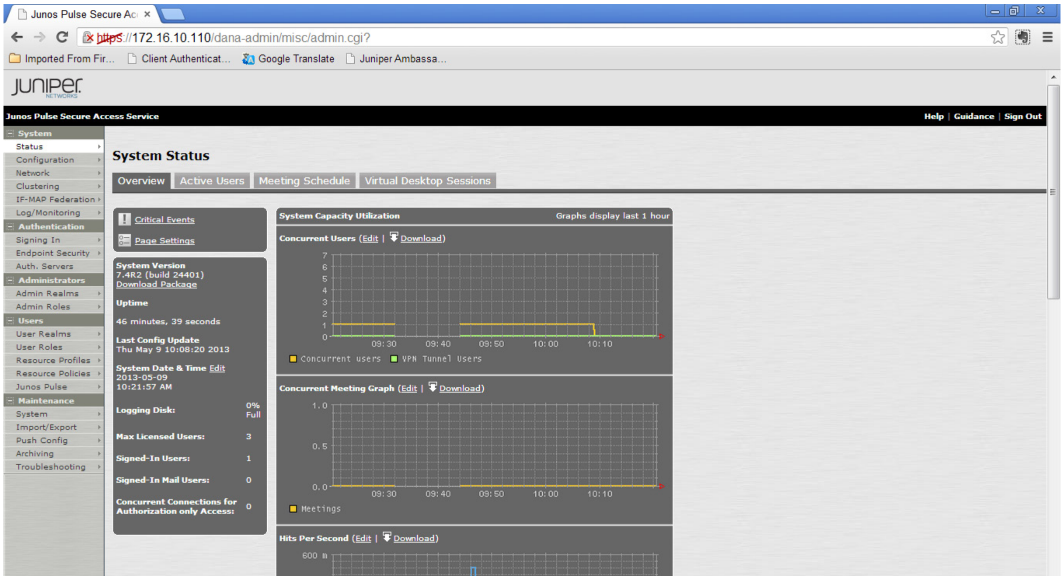
Username
Password

Please sign in to begin your secure session.

Note: This is the **Administrator Sign-In Page**.

If you don't want to sign in as an Administrator, return to the standard Sign-In Page.

2. After logging on, the *System Status* page should be displayed.



System Status

System Version
7.4R2 (build 24401)
Download Package

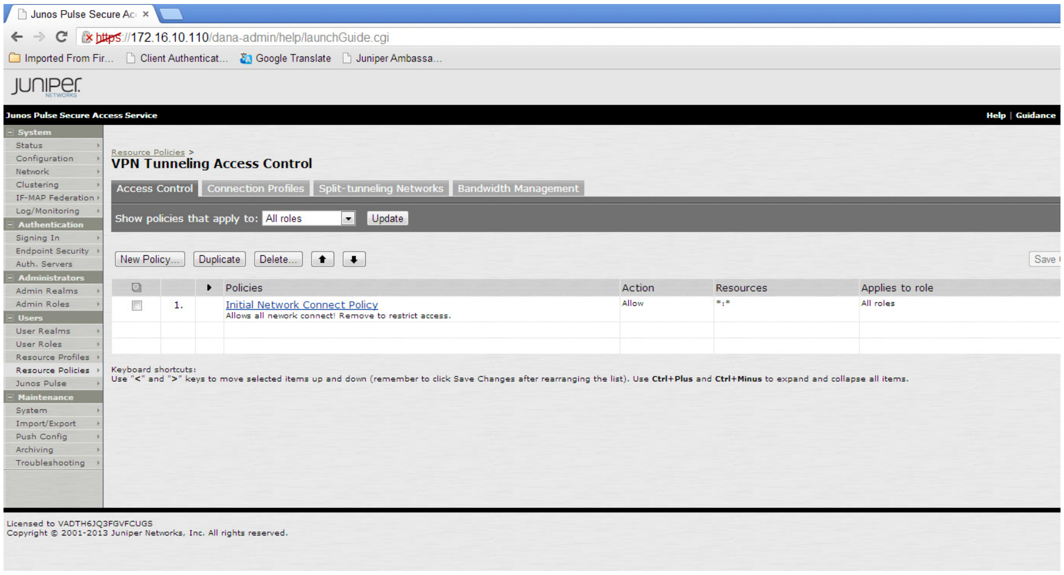
Uptime
46 minutes, 39 seconds

Last Config Update
Thu May 9 10:08:20 2013

System Date & Time Edit
2013-05-09
10:21:57 AM

Logging Disk:	0% Full
Max Licensed Users:	3
Signed-In Users:	1
Signed-In Mail Users:	0
Concurrent Connections for Authorization only Access:	0

3. Navigate in the left-hand margin to *Resource Polices*, then to *VPN Tunneling*, and then to the *Access Control* tab. The default *Access Control* policy permits all access for all authenticated user roles. This is too permissive for most scenarios, so a new policy will be created by clicking the *New Policy...* button.



https://172.16.10.110/dana-admin/help/launchGuide.cgi

Junos Pulse Secure Access Service

Resource Policies >

VPN Tunneling Access Control

Show policies that apply to: All roles

		Policies	Action	Resources	Applies to role
	1.	Initial Network Connect Policy Allows all nework connect! Remove to restrict access.	Allow	*:*	All roles

Keyboard shortcuts:
Use "<" and ">" keys to move selected items up and down (remember to click Save Changes after rearranging the list). Use **Ctrl+Plus** and **Ctrl+Minus** to expand and collapse all items.

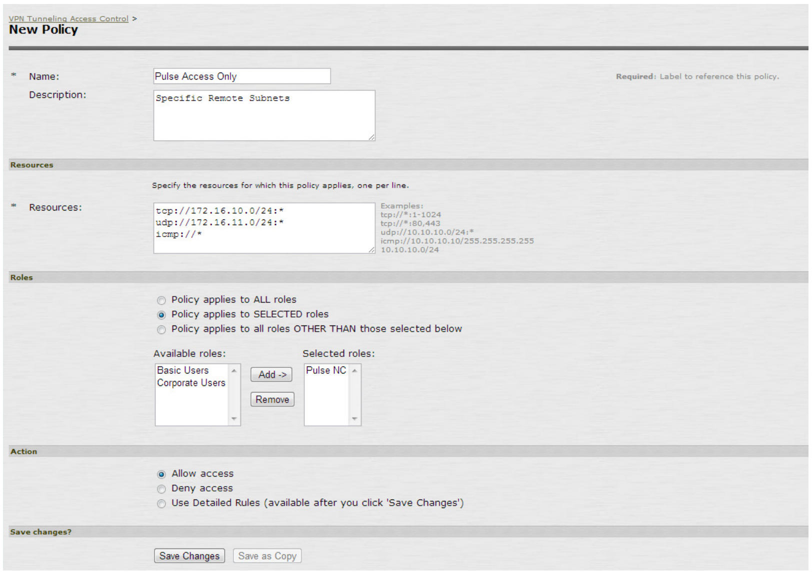
Licensed to VADTH6JQ3FGVFCUGS
Copyright © 2001-2013 Juniper Networks, Inc. All rights reserved.

4. Give the policy a descriptive name such as *Pulse Access Only* and provide some detail in the *Description* field.

5. In the resources field, itemize the protocols (TCP, UDP, ICMP) the resources (either as host IP or subnets) and the ports that they are able to access. For example, the following policy will permit access to all TCP ports on the 172.16.10.0 network, the UDP ports 53 and 1812 on the host 172.16.11.10, and ICMP (Ping) to any other host:

```
tcp://172.16.10.0/24:*
udp://172.16.11.10:53,1812
icmp://*:*
```

6. Scroll down and change the radio button to *Policy applies to SELECTED roles*. Pick a suitable role and add it to the right-hand list. Leave the action as *Allow access* and click the *Save Changes* button.


VPN Tunneling Access Control >

New Policy

* Name:	Pulse Access Only
Description:	Specific Remote Subnets

Required: Label to reference this policy.

Resources

Specify the resources for which this policy applies, one per line.

* Resources:

```
tcp://172.16.10.0/24:*
udp://172.16.11.0/24:*
icmp://*
```

Examples:
```
tcp://*:1-1024
tcp://*:80,443
udp://10.10.10.0/24:*
icmp://10.10.10.10/255.255.255.255
10.10.10.0/24
```

Roles

- Policy applies to ALL roles
- [✓ Policy applies to SELECTED roles]
- Policy applies to all roles OTHER THAN those selected below

Available roles: Basic Users, Corporate Users

Selected roles: Pulse NC

Action

- [✓ Allow access]
- Deny access
- Use Detailed Rules (available after you click 'Save Changes')

Save changes?

7. You should be returned to the *Access Control* menu with the newly created policy listed first. Remember to delete or otherwise disable *Initial Network Connect Policy* — a more restrictive policy may be ineffective.

Creating a Connection Profile

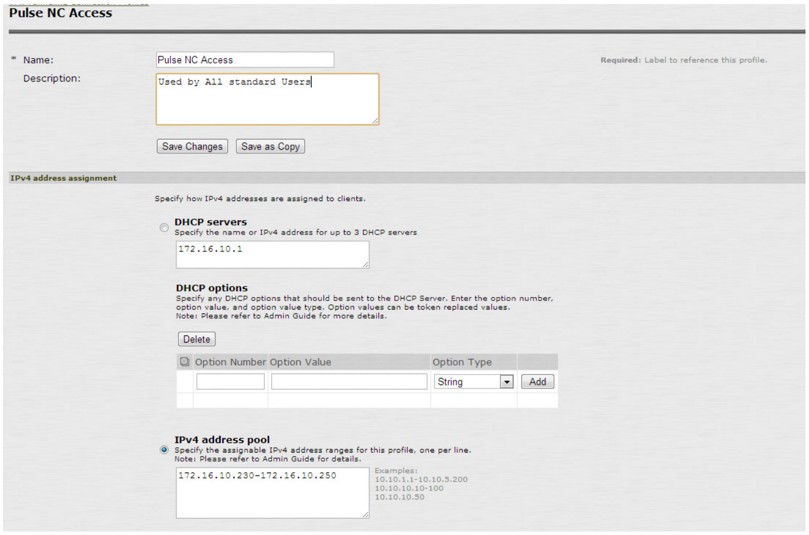
Connection profiles describe how users are assigned an IP address, the VPN tunnel type used, and various other settings such as encryption and proxy handling. In most circumstances, corporate users will share a single *Connection Profile*, however for third-party access you may choose to allocate an alternative IP address range.

1. Navigate to *Resource Policies*, *VPN Tunneling*, and then the *Connection Profiles* tab. Click *New Profile* to create a blank template.

2. Name and describe the profile. The Network Connect and Junos Pulse client can either lease IP addresses from an external DHCP server or the IVE platform itself. If you intend on using an existing server, make sure that the internal interface of the IVE platform is able to lease addresses. Alternatively, enter the range of IP addresses you wish the IVE to allocate to connection clients.

NOTE Make sure that the IP address range allocated is not used by any other network device or DHCP server, or else users will experience intermittent connectivity issues.

NOTE If the IP address range allocated to connecting clients is on the same subnet as the IVE internal interface, the IVE OS will automatically *Proxy ARP* for those IP addresses, requiring no network changes. However, if the IP address range is on a different subnet than the IVE internal interface, network routing changes will be required to direct return traffic into the internal port or cluster VIP.



Pulse NC Access

* Name:	Pulse NC Access	**Required:** Label to reference this profile.
Description:	Used by All standard Users	

IPv4 address assignment

Specify how IPv4 addresses are assigned to clients.

DHCP servers
Specify the name or IPv4 address for up to 3 DHCP servers

```
172.16.10.1
```

DHCP options
Specify any DHCP options that should be sent to the DHCP Server. Enter the option number, option value, and option value type. Option values can be token replaced values.
Note: Please refer to Admin Guide for more details.

	Option Number	Option Value	Option Type	
			String	Add

IPv4 address pool [✓]
Specify the assignable IPv4 address ranges for this profile, one per line.
Note: Please refer to Admin Guide for details.

```
172.16.10.230-172.16.10.250
```

Examples:
10.10.1.1-10.10.5.200
10.10.10.10-100
10.10.10.50

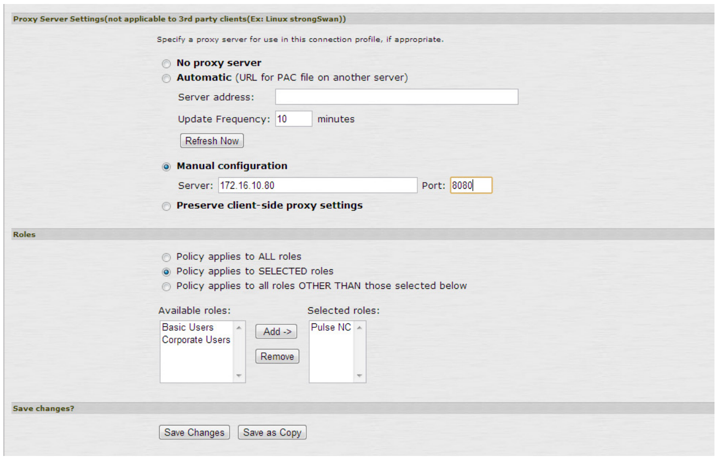
3. Leave the *IPv6* addresses assignment window blank unless you are actively rolling out IPv6 in your network.

4. Scroll down to the *Connection Settings* section. In most cases, the connection should be set to use ESP as the default, but it's worth reading the sidebar for a full discussion on this configuration setting.

5. Scroll down to the *DNS settings* section. By default, Network Connect and Junos Pulse clients will be issued the same DNS servers configured under the *Network* and *Overview* page on the IVE. You can optionally choose to override these settings on a per-connection profile basis, or force the clients to use DHCP server-provided DNS settings.

6. Scroll to the bottom of the window. In the *Proxy Server settings* the browser behavior can be changed to force web requests to use a specific Web Proxy, use a PAC file for automatic configuration, or disable proxy connections entirely.

7. In the *Roles* section, select the *Policies applies to SELECTED roles* radio button and specify which roles this connection profile should apply to, otherwise leave as *Policy applies to ALL roles*. Click Save Change to commit the configuration.



Proxy Server Settings(not applicable to 3rd party clients(Ex: Linux strongSwan))

Specify a proxy server for use in this connection profile, if appropriate.

- ○ **No proxy server**
- ○ **Automatic** (URL for PAC file on another server)
  - Server address:
  - Update Frequency: 10 minutes
  - Refresh Now
- ● **Manual configuration**
  - Server: 172.16.10.80 Port: 8080
- ○ **Preserve client-side proxy settings**

Roles

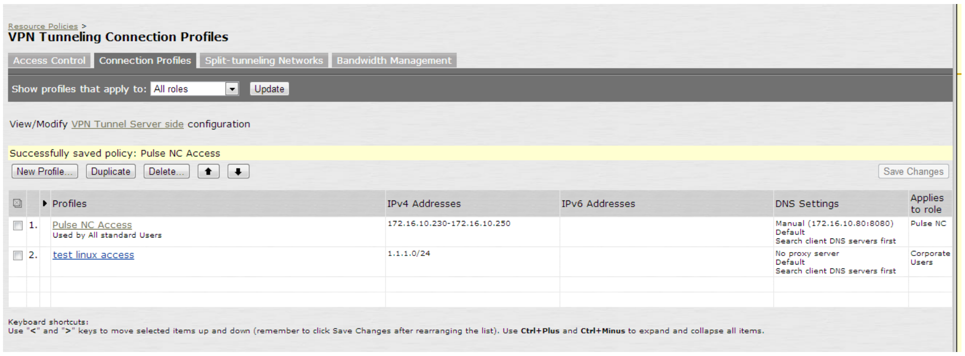
- ○ Policy applies to ALL roles
- ● Policy applies to SELECTED roles
- ○ Policy applies to all roles OTHER THAN those selected below

Available roles: Basic Users, Corporate Users — Add -> / Remove — Selected roles: Pulse NC

Save changes?

Save Changes | Save as Copy

8. You should be returned to the *Connection Profiles* Window. Make sure that each Network Connect and Junos Pulse client enabled role has at least one *Connection Profile* associated with it.



Resource Policies >
VPN Tunneling Connection Profiles

Access Control | Connection Profiles | Split-tunneling Networks | Bandwidth Management

Show profiles that apply to: All roles | Update

View/Modify VPN Tunnel Server side configuration

Successfully saved policy: Pulse NC Access

New Profile... | Duplicate | Delete... | ⬆ | ⬇ | Save Changes

		Profiles	IPv4 Addresses	IPv6 Addresses	DNS Settings	Applies to role
☐	1.	Pulse NC Access Used by All standard Users	172.16.10.230-172.16.10.250		Manual (172.16.10.80:8080) Default Search client DNS servers first	Pulse NC
☐	2.	test linux access	1.1.1.0/24		No proxy server Default Search client DNS servers first	Corporate Users

Keyboard shortcuts:
Use "<" and ">" keys to move selected items up and down (remember to click Save Changes after rearranging the list). Use **Ctrl+Plus** and **Ctrl+Minus** to expand and collapse all items.

Sidebar: Optimize Connectivity with ESP Mode

Both Network Connect and the Junos Pulse Client support ESP failback mode. At the start of the session, user authentication and authorization are conducted over an encrypted SSL connection. Once the user starts Network Connect or Pulse via the portal or stand-alone launcher, a separate point-to-point link is created.

In order to provide the best compromise between performance and compatibility, two distinct tunnel modes are employed: SSL and ESP. In SSL mode the connection is tunneled over a Secure Sockets Layer connection to TCP port 443 on the IVE platform. In ESP mode a tunnel is established using the Encapsulated Security Protocol (bulk encryption algorithm used by IKE) to UDP 4500. Each mode operates more effectively in specific scenarios. For example:

- SSL traffic on TCP 443 is permitted through most firewalls because of its association with encrypted web traffic. However, SSL is relatively CPU intensive on both the client and server, which can result in relatively poor performance.
- ESP is more processor efficient as a protocol than SSL and also allows for a larger packet size; this usually results in higher performance at both ends of the connection. However, ESP is more frequently blocked on firewalls and therefore is less likely to result in a successful connection.

In order to get *the best of both worlds*, when the Network Connect and Pulse client start, they attempt, by default, to initiate a tunnel to the IVE platform on UDP 4500. If this connection fails, after waiting for 15 seconds the client will automatically initiate an SSL connection to the IVE. This approach means that connecting clients will always negotiate the best-possible connection depending on their circumstances. Anecdotally, most connecting clients will either use home-broadband connections or some sort of guest wireless to connect to the IVE. Both of these connection types are typically very permissive of outbound connections and upwards of 90% of connections will negotiate ESP mode. However, there are some circumstances where this automatic negotiation method can actually be a hindrance and some policy tuning is required.

- If UDP 4500 is not permitted through the firewall protecting the IVE platform, every attempt to negotiate ESP mode will fail and result in an unnecessary 15 second wait for each Network Connect or Pulse connection to start. It is recommended that that UDP 4500 is permitted to the external address of the IVE along with the standard ports of HTTP (TCP 80) and HTTPs (TCP 443).
- If a large percentage of clients routinely connect through a restrictive firewall such a third-party corporate network, proxy server, or some types of in-room hotel connections. If large numbers of connecting clients are routinely negotiating SSL mode, despite the IVE firewall permitting it, it is recommended that ESP mode is disabled for these users and SSL-only mode is enforced.
- For regulatory or corporate Policy reasons you require the VPN traffic to be inspected by a non-Juniper device, which cannot decode ESP traffic. In this case it is recommended that you disable ESP mode on the connection profile.

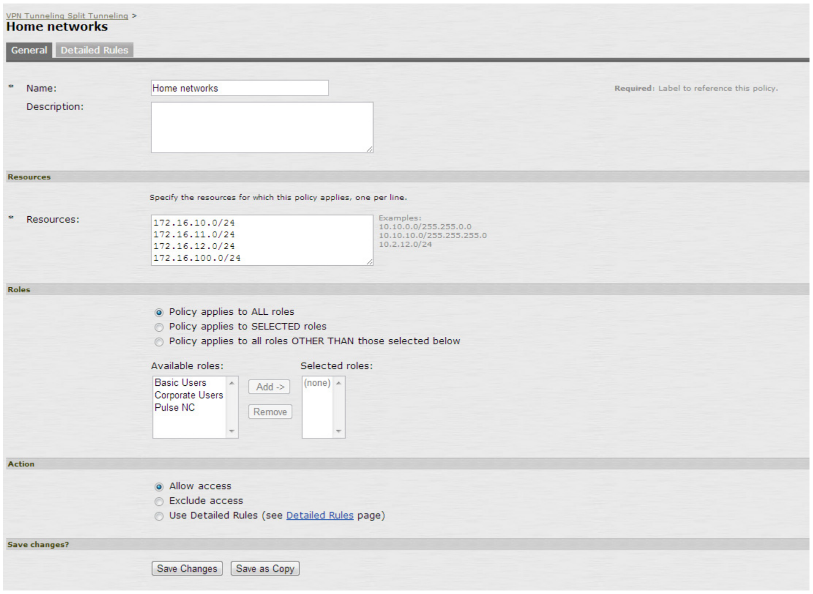
Creating a Split-Tunneling Profile (Optional)

Split tunneling allows the VPN client to simultaneously connect to the corporate network as well as other networks. The most common scenario is to allow users to *surf the web* using their local Internet connection without forcing the traffic down the VPN tunnel to the corporate network. There are many variants that are supported by the IVE platform, but in the example below we will set up a basic *Split-Tunneling* policy that will tunnel access to nominated corporate subnets but allow all other connectivity to route directly out the local gateway. In most cases an organization will only have a single *Split-Tunneling* profile used by all Network Connect and Junos Pulse Client users.

1. Navigate to the *Split-Tunneling* polices by clicking on *Resource Policies*, *VPN Tunneling*, and then the *Split-Tunneling* tab. Click *New Policy…* to create a new blank policy.

2. At the top of the screen name and describe the policy for future reference. In the *Resources* section, list all the corporate networks you wish the Network Connect or Pulse client to route down the VPN tunnel. It is generally recommend that the largest reasonable subnets are used to capture the traffic.

NOTE This policy should not be used to provide fine-grain control over connecting clients as this is the job of the Access policy discussed earlier in this recipe.



Home networks

* Name: Home networks — **Required:** Label to reference this policy.

Description:

Resources

Specify the resources for which this policy applies, one per line.

* Resources:

```
172.16.10.0/24
172.16.11.0/24
172.16.12.0/24
172.16.100.0/24
```

Examples:
10.10.0.0/255.255.0.0
10.10.10.0/255.255.255.0
10.2.12.0/24

Roles

[✓ Policy applies to ALL roles]
Policy applies to SELECTED roles
Policy applies to all roles OTHER THAN those selected below

Available roles: Basic Users, Corporate Users, Pulse NC
Selected roles: (none)

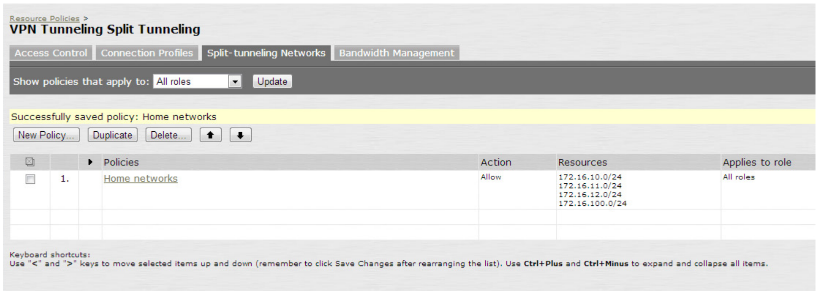
Action

[✓ Allow access]
Exclude access
Use Detailed Rules (see Detailed Rules page)

Save changes?

3. Unless you wish to apply this policy to a specific role, leave the selection as *Policy applies to ALL roles* and click Save *Changes* to commit the configuration.

4. This should return the browser to the *Split-Tunneling* menu. Further configuration is required before split tunneling can be enabled, and this is covered in a later section.



Show policies that apply to: All roles

Successfully saved policy: Home networks

		Policies	Action	Resources	Applies to role
	1.	Home networks	Allow	172.16.10.0/24 172.16.11.0/24 172.16.12.0/24 172.16.100.0/24	All roles

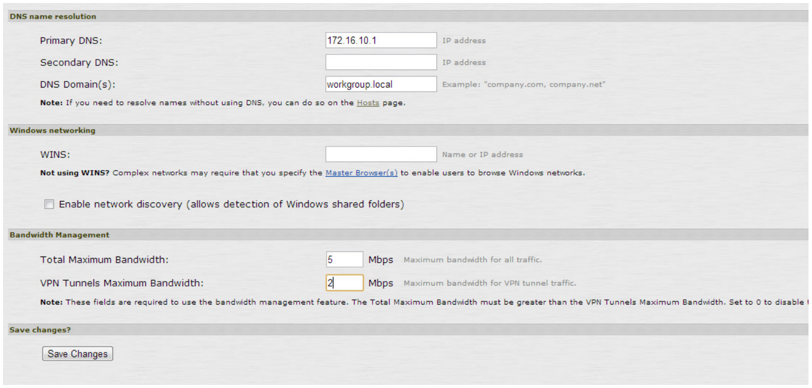
Keyboard shortcuts:
Use "<" and ">" keys to move selected items up and down (remember to click Save Changes after rearranging the list). Use **Ctrl+Plus** and **Ctrl+Minus** to expand and collapse all items.

Creating a Bandwidth Management Policy (Optional)

The IVE platform is able to create a policy that prioritizes access to network connect bandwidth on a per-role basis. This particular policy method is very rarely deployed as providing strict bandwidth controls, and should be used with caution as it can negatively impact performance for all users. Scenarios where Bandwidth management may be considered include:

- Deployments with many thousands of concurrent connections
- Deployments where large volumes of data are routinely transferred in a single session (i.e. files larger than 1 GB)
- Deployments with a large disparity between the client and server-side bandwidth availability (i.e. users with Fiber optic broadband connecting to a 2Mbps leased line circuit)
- A scenario with Premium Users who must always have priority access to network bandwidth
- Any of the above deployments without any upstream-bandwidth management or client-side WAN optimization available.

1. Navigate to the *Network Settings* page and then the *Overview* Tab. Scroll down to *Bandwidth Management* section and enter the values for the *Total Maximum Bandwidth* you wish to be used by the IVE Platform and *VPN Tunnels Maximum Bandwidth* used by Network Connect and the Pulse Client. Click Save Changes to commit the configuration.



DNS name resolution

Primary DNS: 172.16.10.1 IP address

Secondary DNS: IP address

DNS Domain(s): workgroup.local Example: "company.com, company.net"

Note: If you need to resolve names without using DNS, you can do so on the Hosts page.

Windows networking

WINS: Name or IP address

Not using WINS? Complex networks may require that you specify the Master Browser(s) to enable users to browse Windows networks.

☐ Enable network discovery (allows detection of Windows shared folders)

Bandwidth Management

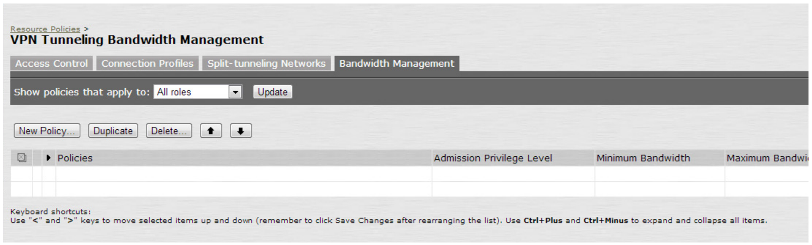
Total Maximum Bandwidth: 5 Mbps Maximum bandwidth for all traffic.

VPN Tunnels Maximum Bandwidth: 2 Mbps Maximum bandwidth for VPN tunnel traffic.

Note: These fields are required to use the bandwidth management feature. The Total Maximum Bandwidth must be greater than the VPN Tunnels Maximum Bandwidth. Set to 0 to disable

Save changes?

2. Navigate to *Resource Policies*, *VPN Tunneling* and then *Bandwidth Management*. Click *New Policy*.



Resource Policies >

VPN Tunneling Bandwidth Management

Show policies that apply to: All roles

Policies	Admission Privilege Level	Minimum Bandwidth	Maximum Bandwi

Keyboard shortcuts:
Use "<" and ">" keys to move selected items up and down (remember to click Save Changes after rearranging the list). Use **Ctrl+Plus** and **Ctrl+Minus** to expand and collapse all items.

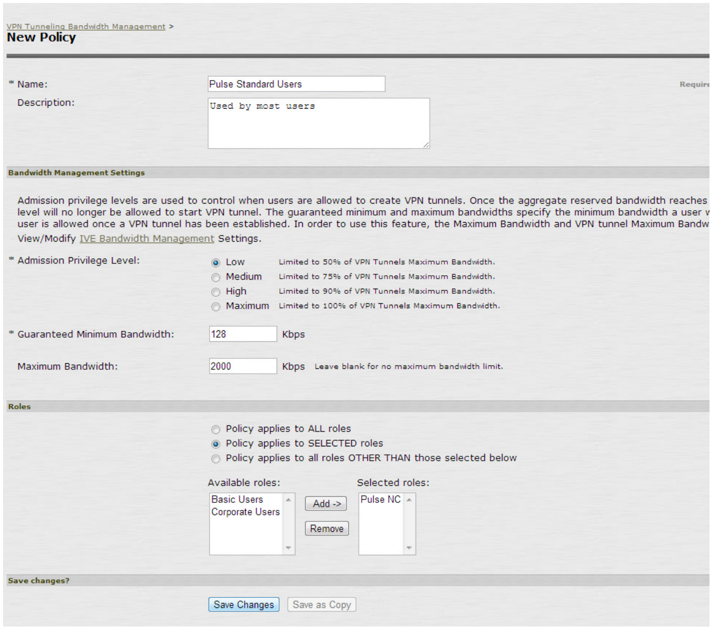
3. Name and describe the policy. Depending on the perceived priority for the user role, select the *Admission Privilege Level*.

NOTE It is recommended that *Maximum* is only used for System Administrators who may need to log on to the appliance via Pulse or Network Connect in order to manually manage user concurrency.

Specify the *Guaranteed Minimum Bandwidth* in Kbps and if applicable the maximum bandwidth utilization for a user who has matched this role.

Select the Roles to which this policy should apply and click *Save Changes* to commit the configuration.

WARNING! Consider the *Guaranteed Minimum Bandwidth* and *Maximum Bandwidth* settings very carefully, as they are applied to each user within the associated rule. As a result it is very easy to over-commit the available bandwidth with only a few concurrent connections and impact performance for all remaining users.



VPN Tunneling Bandwidth Management >
New Policy

* Name:	Pulse Standard Users
Description:	Used by most users

Bandwidth Management Settings

Admission privilege levels are used to control when users are allowed to create VPN tunnels. Once the aggregate reserved bandwidth reaches level will no longer be allowed to start VPN tunnel. The guaranteed minimum and maximum bandwidths specify the minimum bandwidth a user v user is allowed once a VPN tunnel has been established. In order to use this feature, the Maximum Bandwidth and VPN tunnel Maximum Bandw View/Modify IVE Bandwidth Management Settings.

* Admission Privilege Level:

- [✓ Low] Limited to 50% of VPN Tunnels Maximum Bandwidth.
- Medium Limited to 75% of VPN Tunnels Maximum Bandwidth.
- High Limited to 90% of VPN Tunnels Maximum Bandwidth.
- Maximum Limited to 100% of VPN Tunnels Maximum Bandwidth.

* Guaranteed Minimum Bandwidth: 128 Kbps

Maximum Bandwidth: 2000 Kbps Leave blank for no maximum bandwidth limit.

Roles

- Policy applies to ALL roles
- [✓ Policy applies to SELECTED roles]
- Policy applies to all roles OTHER THAN those selected below

Available roles: Basic Users, Corporate Users

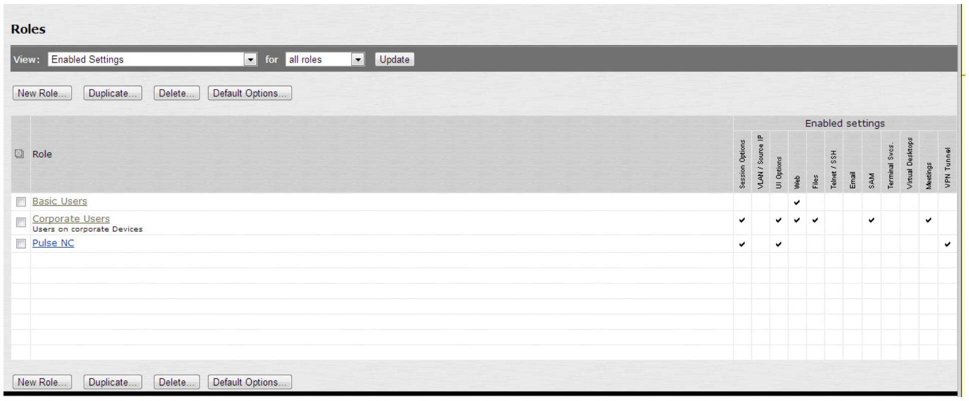
Selected roles: Pulse NC

Save changes?

Configuring VPN Tunneling

Now that the prerequisite Network Connect and Junos Pulse clients configurations have been created, some further per-role tuning can be configured.

1. Navigate to User Roles, this will show all configured roles on the system and those that have VPN tunneling enabled. Either create a new Role and enable VPN Tunneling or edit an existing one by clicking on it.

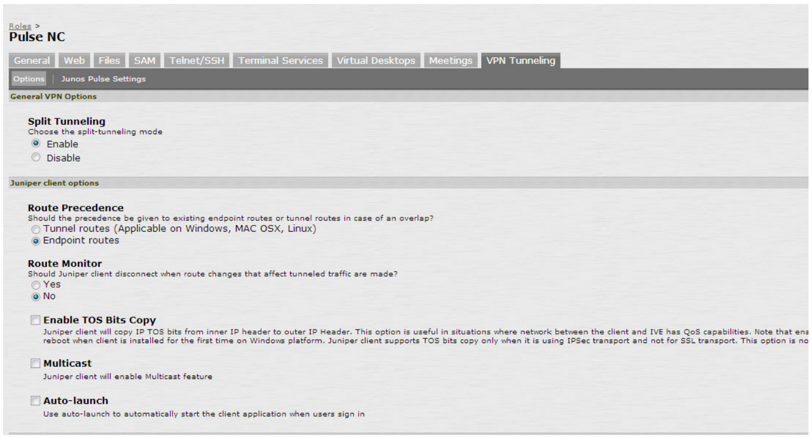


Roles

View: Enabled Settings for all roles

	Enabled settings											
Role	Session Options	VLAN / Source IP	UI Options	Web	Files	Telnet / SSH	Email	SAM	Terminal Svcs.	Virtual Desktops	Meetings	VPN Tunnel
Basic Users				✓								
Corporate Users Users on corporate Devices	✓		✓	✓	✓			✓			✓	
Pulse NC	✓		✓									✓

2. Navigate to the *VPN Tunneling* tab and then to the *Options* tab. Enable Split Tunneling to allow the connecting client to simultaneously enable access to the corporate network and the Internet.


Roles >
Pulse NC

General | Web | Files | SAM | Telnet/SSH | Terminal Services | Virtual Desktops | Meetings | VPN Tunneling

Options | Junos Pulse Settings

General VPN Options

Split Tunneling
Choose the split-tunneling mode
[✓ Enable] Disable

Juniper client options

Route Precedence
Should the precedence be given to existing endpoint routes or tunnel routes in case of an overlap?
Tunnel routes (Applicable on Windows, MAC OSX, Linux) [✓ Endpoint routes]

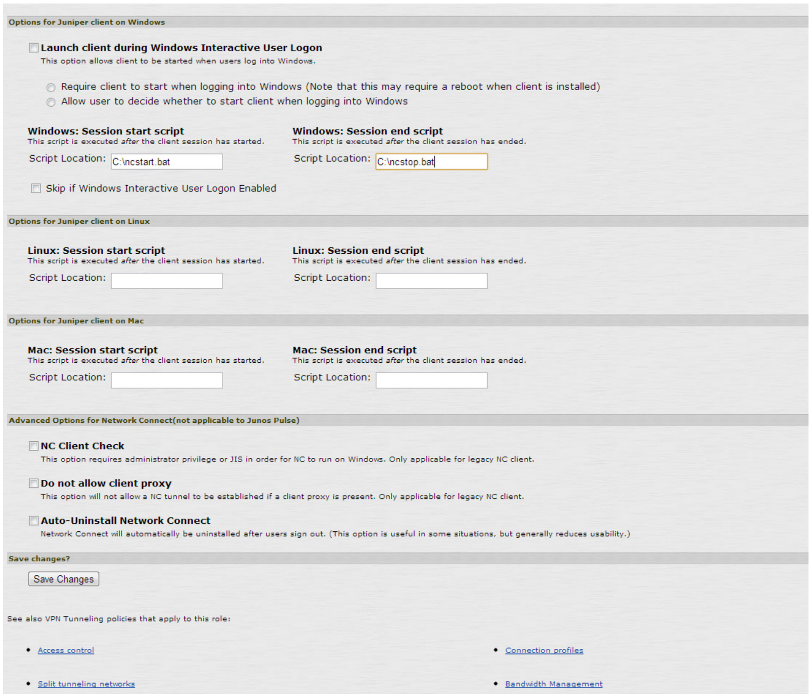
Route Monitor
Should Juniper client disconnect when route changes that affect tunneled traffic are made?
Yes [✓ No]

☐ **Enable TOS Bits Copy**
Juniper client will copy IP TOS bits from inner IP header to outer IP Header. This option is useful in situations where network between the client and IVE has QoS capabilities. Note that ena
reboot when client is installed for the first time on Windows platform. Juniper client supports TOS bits copy only when it is using IPSec transport and not for SSL transport. This option is no

☐ **Multicast**
Juniper client will enable Multicast feature

☐ **Auto-launch**
Use auto-launch to automatically start the client application when users sign in

3. Scroll down to *Options for Juniper client on Windows*. This section allows you to specify script files (MS DOS Batch) to be executed on the connecting client after VPN tunnel establishment and after termination. This provides the opportunity to map network drives, force *Group Policy* updates, and update client components, etc. Similar options are also available for the Linux and Mac Platforms. Click *Save Changes* to commit the configuration.


Options for Juniper client on Windows

☐ **Launch client during Windows Interactive User Logon**
This option allows client to be started when users log into Windows.

Require client to start when logging into Windows (Note that this may require a reboot when client is installed)
Allow user to decide whether to start client when logging into Windows

Windows: Session start script
This script is executed *after* the client session has started.
Script Location: C:\ncstart.bat

Windows: Session end script
This script is executed *after* the client session has ended.
Script Location: C:\ncstop.bat

☐ Skip if Windows Interactive User Logon Enabled

Options for Juniper client on Linux

Linux: Session start script
This script is executed *after* the client session has started.
Script Location:

Linux: Session end script
This script is executed *after* the client session has ended.
Script Location:

Options for Juniper client on Mac

Mac: Session start script
This script is executed *after* the client session has started.
Script Location:

Mac: Session end script
This script is executed *after* the client session has ended.
Script Location:

Advanced Options for Network Connect(not applicable to Junos Pulse)

☐ **NC Client Check**
This option requires administrator privilege or JIS in order for NC to run on Windows. Only applicable for legacy NC client.

☐ **Do not allow client proxy**
This option will not allow a NC tunnel to be established if a client proxy is present. Only applicable for legacy NC client.

☐ **Auto-Uninstall Network Connect**
Network Connect will automatically be uninstalled after users sign out. (This option is useful in some situations, but generally reduces usability.)

Save changes?

Save Changes

See also VPN Tunneling policies that apply to this role:

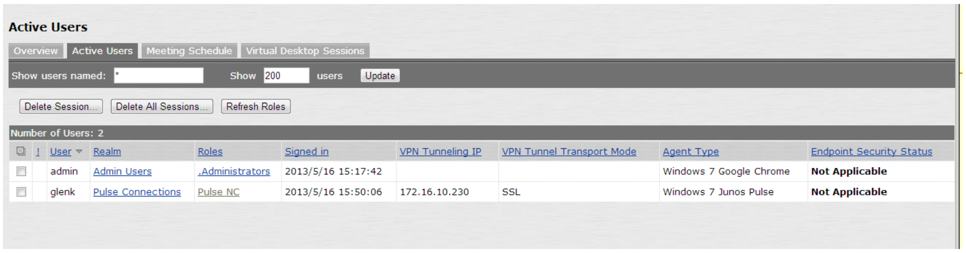
- Access control
- Connection profiles
- Split tunneling networks
- Bandwidth Management

Testing the Policies

Now that the policies and roles have been enabled, it is possible to launch either Network Connect or the Junos Pulse client and review the roles and access that have been received.

1. Navigate to *Status* and *Active Users*. This will show all users presently logged in to the IVE platform. This should include the following settings of interest:

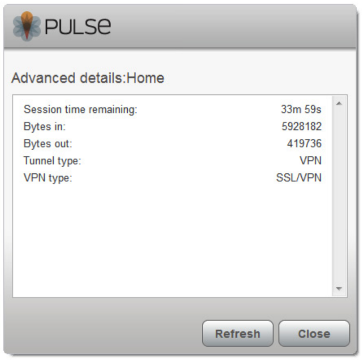
- User: Successfully authenticated primary username
- Realm: The realm the user logged in to
- Roles: All roles that the user has successfully been granted access to
- VPN Tunneling IP: IP addresses issued to the connecting client
- VPN Tunnel Transport Mode: Whether the user negotiated either SSL or ESP mode
- Agent Type: Platform and agent of connecting client
- Endpoint Security Status: Outcome of any host check tests (if enabled)



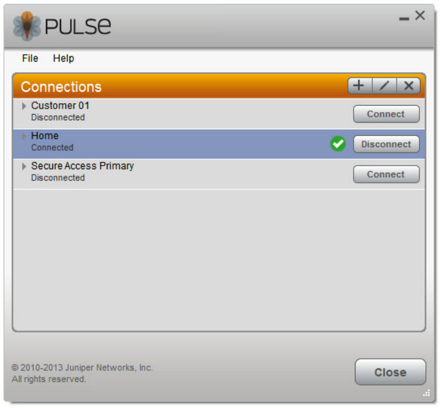
Number of Users: 2

!	User	Realm	Roles	Signed in	VPN Tunneling IP	VPN Tunnel Transport Mode	Agent Type	Endpoint Security Status
	admin	Admin Users	.Administrators	2013/5/16 15:17:42			Windows 7 Google Chrome	**Not Applicable**
	glenk	Pulse Connections	Pulse NC	2013/5/16 15:50:06	172.16.10.230	SSL	Windows 7 Junos Pulse	**Not Applicable**

2. Open the connecting client and view Advanced Details for the connection. This should be the same information as above.



Session time remaining:	33m 59s
Bytes in:	5928182
Bytes out:	419736
Tunnel type:	VPN
VPN type:	SSL/VPN



Recipe 11

Creating a Simple Junos Op Script

The Junos Operating System is extremely powerful and flexible, not to mention its intelligence and numerous features. More to the point, Junos allows you to customize using *scripting*, which gives you the ability to perform such tasks as creating your own show commands and configuration macros.

MORE? This recipe guides you through creating your first script. It is beyond the scope of the recipe to provide details on the numerous options and possible alternative solutions available, so for further study, see the *Resources* section at the end of this recipe.

Discussion

In this recipe, you will create a script that issues several operational-mode commands and writes output to the display. The idea is to create a customizable version of a Junos "request support information" command that you will be able to easily expand. To start, you will create a script that issues three commands:

```
show chassis hardware detail
show version
show system storage
```

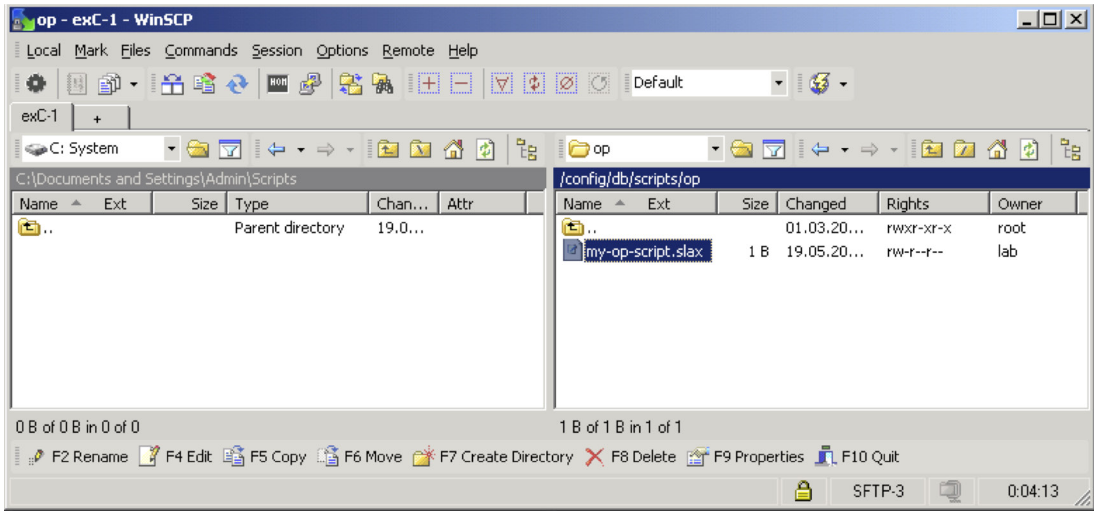
First of all, there are two programming languages in which you can write Junos scripts: XLST and SLAX. XSLT is standards-based, while SLAX is an alternative open-source language developed by Juniper and is actually a pre-processor to XSLT. Because the syntax of SLAX is more easily understandable to engineers that have worked with such traditional languages as C/C++, Pascal, Perl, or Java, let's use SLAX here.

There are three types of Junos scripts: operational (op) scripts, commit scripts, and event scripts. Op scripts are the easiest to understand – they might be considered as a custom Junos command. However, op scripts can perform much more than just putting some output to the screen. An op script can change configuration, for example, but this recipe needs to constrain itself with a more simple application.

MORE? Scripts work on any Junos box, be it a router, a switch, or a security device, and do not require any license.

To do their job, op scripts must be put in the following directory on the Junos device: */var/db/scripts/op*. You can create a script with any text editor you like and then copy it to the device using any application, such as WinSCP (download here: http://winscp.net/eng/download.php).

NOTE WinSCP is not provided by Juniper Networks. It is just a convenient third-party open-source SCP/FTP client with a built-in text editor that can be used to edit Junos script files.


op - exC-1 - WinSCP
Local Mark Files Commands Session Options Remote Help
Default
exC-1
C: System
op
F2 Rename F4 Edit F5 Copy F6 Move F7 Create Directory F8 Delete F9 Properties F10 Quit
SFTP-3
0:04:13

C:\Documents and Settings\Admin\Scripts

```
Name  Ext   Size  Type              Chan...  Attr
..                Parent directory  19.0...
```

0 B of 0 B in 0 of 0

/config/db/scripts/op

```
Name              Ext  Size  Changed     Rights      Owner
..                           01.03.20... rwxr-xr-x   root
my-op-script.slax      1 B   19.05.20... rw-r--r--   lab
```

1 B of 1 B in 1 of 1

Figure 11.1 WinSCP Application Screen

In Figure 11.1, you are logged in to a Junos device (an EX4200 switch, in this case) with WinSCP using SCP protocol (it allows secure file transfer via SSH channel, usually using standard SSH port number

22). In the left panel is your local PC or laptop, in the right panel is the Junos filesystem. Navigate to /var/db/scripts/op and create a new file *my-op-script.slax* (to do this, right-click and select New -> File). Double-click the newly created file to edit it directly on the box.

NOTE Do not forget to save the file every time you want to run the edited script. Use the Save button or press Ctrl-S on your keyboard.

Here is how your script *my-op-script.slax* may look to perform the tasks outlined in the "Problem" section. The lines are numbered for further referencing.

```
(01) version 1.0;
(02) ns junos = "http://xml.juniper.net/junos/*/junos";
(03) ns xnm = "http://xml.juniper.net/xnm/1.1/xnm";
(04) ns jcs = "http://xml.juniper.net/junos/commit-scripts/1.0";
(05)
(06) import "../import/junos.xsl";
(07)
(08) match / {
(09)   <op-script-results> {
(10)
(11)     <output> "***** Chassis hardware: *****";
(12)     var $cmd1 = <command> 'show chassis hardware detail';
(13)     copy-of (jcs:invoke($cmd1));
(14)
(15)     <output>;
(16)     <output> "***** Version: *****";
(17)     var $cmd2 = <command> 'show version';
(18)     copy-of (jcs:invoke($cmd2));
(19)
(20)     <output>;
(21)     <output> "***** System storage: *****";
(22)     var $cmd3 = <command> 'show system storage';
(23)     copy-of (jcs:invoke($cmd3));
(24)   }
(25) }
```

Lines (**01-04**) of the script are a recommended header that are likely the same for most of Junos scripts, as they define the SLAX version and commonly used namespaces.

Line (**06**) imports a file with several very useful templates (subroutines) that you can use while writing your script. Although you may skip this line in our example script, this is part of a recommended header, so it is wise to leave it as is.

Line (**08**) is a start of the main template of the script, defining its entry point. Junos scripts are based on XML, and every script is actually working with two XML documents: input document and output one. The `match /` just means that it starts our script once the root node (`/`) of the input XML document is found.

Line (**09**) encloses the script's future output in <op-script-results> XML tags, which is needed for Junos to understand it.

Line (**11**) writes to the outgoing XML document the `***** Chassis hardware: *****` line enclosed in opening `<output>` and closing `</output>` XML tags. Due to this, Junos CLI writes this string to the console, as you will see below.

Line (**12**) creates a variable and assigns it a value of XML RPC (remote procedure call) corresponding to the `show chassis hardware detail` command. You use a tag `<command>`, which is the easiest way to create the needed content for the variable.

NOTE An alternative way to get the required XML RPC for the command would be to issue the following Junos command:

```
show chassis hardware detail | display xml rpc
```

... and assign the corresponding XML output to `$cmd1` variable. See the resources referenced at the end of this recipe for more details.

Line (13) uses the function `jcs:invoke()` to actually run the command corresponding to `$cmd1` variable (i.e., `show chassis hardware detail`). Its output is redirected to the user (output XML document) with copy-of statement.

Line (**15**) creates an empty line.

Further code basically repeats lines (11-13) for the other two commands that this recipe is using the script to issue.

After you enter this script and save it (or copy the file from a remote location to /var/db/scripts/op), you need to allow Junos to run the script by using configuration-mode commands:

```
{master:0}[edit]
lab@exC-1# set system scripts op file my-op-script.slax

{master:0}[edit]
lab@exC-1# commit
```

Now run the script with op <filename> command and see the output (some of it is skipped to save space):

```
lab@exC-1> op my-op-script
***** Chassis hardware: *****
Hardware inventory:
Item             Version  Part number  Serial number     Description
Chassis                                BM0291313274      EX4200-24T
Routing Engine 0 REV 13   750-033065   BM0291313274      EX4200-24T, 8 POE

...
```

```
***** Version: *****
fpc0:
--------------------------------------------------------------------------
Hostname: exC-1
Model: ex4200-24t
JUNOS Base OS boot [11.4R7.5]

...

***** System storage: *****

fpc0:
--------------------------------------------------------------------------
Filesystem              Size       Used      Avail  Capacity   Mounted on
/dev/da0s2a             183M       100M        68M       60%  /
devfs                   1.0K       1.0K         0B      100%  /dev
...
```

As you can see, the output of all three commands is displayed, with only one operator's command – so automation is indeed happening as promised.

NOTE If you would like to see the script's XML output (i.e., the output XML document that is created by the script and sent to Junos CLI for processing), issue the command:

```
op my-op-script | display xml
```

And it is one of several possible methods to troubleshoot op scripts.

Summary

You may have noticed that Junos configuration looks more like a programming language than just a list of settings. Junos scripts are a method to go even further with Junos, creating your own commands and adding more intelligence in the way that *you* want.

Resources

Junos as a Scripting Language – this is an online training course provided by Juniper:

https://learningportal.juniper.net/juniper/user_activity_info.aspx?id=3328

This Week: Applying Junos Automation – this book, by Curtis Call, is a free download for J-Net members:

https://www.juniper.net/us/en/community/junos/training-certification/day-one/automation-series/applying-junos-automation/

This Week: Mastering Junos Automation Programming – this book by Jeremy Schulman and Curtis Call is free to all J-Net members:

http://www.juniper.net/us/en/community/junos/training-certification/day-one/automation-series/mastering-junos-automation/

For extensive coverage of all available Junos functions and options see the Junos technical documentation: *Configuration and Operations Automation Guide*. It is renewed with every Junos release, so use the one corresponding to your Junos version, so for Junos 12.1:

http://www.juniper.net/techpubs/en_US/junos12.1/information-products/topic-collections/config-guide-automation/config-guide-automation.pdf

Another useful source of information is the Junos Script Library, which contains many example scripts that you can use for work and study:

http://www.juniper.net/us/en/community/junos/script-automation/library/

Recipe 12

Destination NAT in a Dual ISP Environment

It is common for branch offices to have two Internet connections for redundancy. Such connections are used for branch user traffic to the Internet as well as for providing access to some internal resources, such as servers, located in the internal network (or DMZ) behind a firewall. For providing access to such internal resources that don't have a public IP address, destination NAT (DNAT) is commonly used. Although configuring DNAT with one ISP connection is a relatively easy task, a dual ISP environment requires a little special treatment.

Problem

As shown in Figure 12.1, the SRX has two Internet connections: ISP1 on interface ge-0/0/1; and, ISP2 on interface ge-0/0/2. The remote host needs to be able to initiate telnet connections to the "Internal host" server protected by the SRX firewall (in this setup, we'll test with IP address 7.7.7.7 for the remote host, but the same should work with any remote IP).

NOTE The lab's setup uses telnet (TCP port 23) as a service that is provided by an internal host, however, the configuration will be very similar for any other service such as HTTP, HTTPS, or SSH.

Because the internal host has a private IP address (192.168.50.50), DNAT must be used to access it from the Internet.

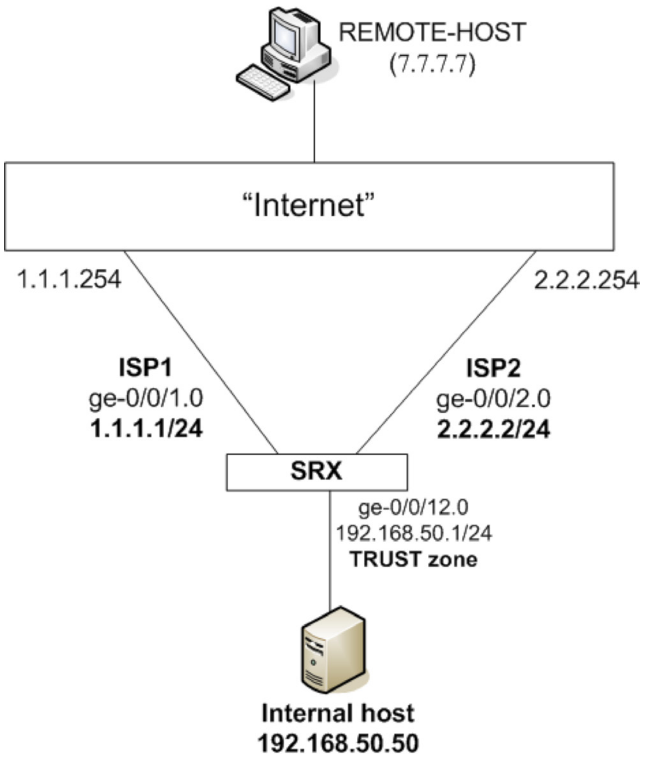


Figure 12.1 Dual ISP Test Lab Setup

NOTE Interface addresses 1.1.1.1 and 2.2.2.2 are used for providing access to the internal server, because it is a common situation that ISPs provide only one public IP address to their customers.

The task is completed once you are able to telnet to 1.1.1.1 and 2.2.2.2 from 7.7.7.7 and get access to the internal host.

CAUTION You should be careful when opening access to your internal resources in such a manner and make sure non-legitimate users don't get access to your data. One possible way to achieve this is to make sure that external users are properly authenticated on the internal server before they get access to any sensitive resources. In addition, you can decide to allow only a limited number of source addresses to enter your network by means of security policy.

Solution

Here is the configuration of interfaces, security zones, and routing:

```
lab@SRX# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet {
            address 1.1.1.1/24;
        }
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 2.2.2.2/24;
        }
    }
}
ge-0/0/12 {
    unit 0 {
        family inet {
            address 192.168.50.1/24;
        }
    }
}

[edit]
lab@SRX# show security zones
security-zone ISP1 {
    host-inbound-traffic {
        system-services {
            ping;
        }
    }
    interfaces {
        ge-0/0/1.0;
    }
}
security-zone ISP2 {
    host-inbound-traffic {
        system-services {
            ping;
        }
    }
    interfaces {
        ge-0/0/2.0;
    }
}
security-zone TRUST {
    address-book {
        address Internal-host 192.168.50.50/32;
    }
    interfaces {
        ge-0/0/12.0;
    }
}
static {
    route 0.0.0.0/0 {
        next-hop 1.1.1.254;
        qualified-next-hop 2.2.2.254 {
            preference 7;
        }
    }
}
```

NOTE The TRUST zone has an entry for internal host in the address book configured. This will be used for security policy configuration.

NOTE There are two ISPs, so the default static route has two next hops – primary 1.1.1.254 for ISP1 (this route has a default preference value of 5) and secondary 2.2.2.254 for ISP2 (for this route, preference value of 7 is configured using the **qualified-next-hop** knob, so it only becomes active when link to ISP1 is down).

Now add a destination NAT configuration and policy to allow telnet traffic that is coming in (initially to 1.1.1.1 and/or 2.2.2.2, but the destination IP for each session will be translated to 192.168.50.50):

```
lab@SRX# show security nat destination
pool INT-HOST {
    address 192.168.50.50/32;
}
rule-set ISP1 {
    from zone ISP1;
    rule dnat-isp-1 {
        match {
            destination-address 1.1.1.1/32;
            destination-port 23;
        }
        then {
            destination-nat pool INT-HOST;
        }
    }
}
rule-set ISP2 {
    from zone ISP2;
    rule dnat-isp-2 {
        match {
            destination-address 2.2.2.2/32;
            destination-port 23;
        }
        then {
            destination-nat pool INT-HOST;
        }
    }
}
```

```
[edit]
lab@SRX# show security policies
from-zone ISP1 to-zone TRUST {
    policy DNAT1 {
        match {
            source-address any;
            destination-address Internal-host;
            application junos-telnet;
        }
        then {
            permit;
        }
    }
}
from-zone ISP2 to-zone TRUST {
    policy DNAT2 {
        match {
            source-address any;
            destination-address Internal-host;
            application junos-telnet;
        }
        then {
            permit;
        }
    }
}
```

Note the DNAT pool with the address of the internal host in it. Also configured were two DNAT rules that match on traffic for destination port 23 (telnet) coming to 1.1.1.1 and 2.2.2.2, respectively, and do DNAT to that pool. Finally, security policies are configured – remember that policies must match on already translated destination addresses. Let's commit to apply the configuration changes, and check the result by doing telnet from the remote host:

```
lab@REMOTE-HOST> telnet 1.1.1.1
Trying 1.1.1.1...
Connected to 1.1.1.1.
Escape character is '^]'.

INT-HOST (ttyp2)

login: ^C Client aborted login
Connection closed by foreign host.

lab@REMOTE-HOST> telnet 2.2.2.2
Trying 2.2.2.2...
telnet: connect to address 2.2.2.2: Operation timed out
telnet: Unable to connect to remote host
```

You can see that that first DNAT rule works fine (telnet session opens and it's closed with Control-C), but there is a problem with telnetting to the second ISP's IP address (session on remote host times out). Let's investigate the issue by viewing the session table on the SRX right after trying to initiate the session to 2.2.2.2:

```
lab@SRX> show security flow session destination-prefix 2.2.2.2
Session ID: 39564, Policy name: DNAT2/5, Timeout: 12, Valid
  In: 7.7.7.7/52514 --> 2.2.2.2/23;tcp, If: ge-0/0/2.0, Pkts: 3, Bytes: 192
  Out: 192.168.50.50/23 --> 7.7.7.7/52514;tcp, If: ge-0/0/12.0, Pkts: 0, Bytes: 0
Total sessions: 1
```

As you can see, packets are coming in (bold), but not going out. Our task now is to understand the problem and fix it.

TRY THIS The answer is coming shortly, but can you guess what the problem is at this point?

First of all, DNAT and security policy seem to work because there is an entry in the session table. Although it may not seem obvious at first glance, the problem is in routing. Every time SRX sets up a session, it checks a route to the destination, as well as the reverse route to the source of traffic. This reverse route lookup is needed so that SRX knows where to send packets back for a return flow of a session. For our REMOTE-HOST initiating a telnet session to 2.2.2.2, the reverse route lookup (which is made for session's source address, 7.7.7.7) points to the interface of ISP1, because the active default route points to 1.1.1.254.

So, for a session initiated by the remote host to the address 2.2.2.2 of ISP2, the return traffic will be sent by the SRX via ISP1. In our case, traffic is dropped because ISP1 and ISP2 are in different zones. To confirm this, let's check the flow traceoptions:

```
lab@SRX# show security flow
traceoptions {
    file flow-log;
    flag packet-drops;
    flag basic-datapath;
    packet-filter filter-dnat {
        destination-prefix 2.2.2.2/32;
        destination-port 23;
    }
    packet-filter filter-dnat-reverse {
        destination-prefix 7.7.7.7/32;
        source-port 23;
    }
}
```

And if you tried to initiate traffic (telnet 2.2.2.2 from 7.7.7.7) once again, you'd see something like the following in the file flow-log:

```
Apr 13 14:12:05 14:12:05.698794:CID-0:RT:  route lookup failed: dest-
ip 7.7.7.7 orig ifp       ge-0/0/2.0 output_ifp ge-0/0/1.0 fto 0x459f7cb0 orig-
zone 8 out-zone 7 vsd 0
Apr 13 14:12:05 14:12:05.698794:CID-0:RT:  packet dropped, pak dropped since re-
route failed
```

Indeed, the return flow of the session tries to come back via ge-0/0/1.0 and traffic is dropped because the interfaces belong to different security zones. If you move both ISP uplinks to one zone and modify the NAT configuration accordingly, the traffic will (at least in lab setup) start flowing, but such asymmetric routing (i.e., traffic coming in via ISP2 and leaving via ISP1) is not what is desired (it will likely be dropped on its way by ISP1 *and anyway* it does not provide the redundancy in our setup).

The solution to the problem is to put ISP2's interface in a separate routing instance, of a virtual router type (*vrouter*, for short), on the SRX. In this case, for traffic coming in via ISP2, the reverse route lookup will be done using this virtual router's routing table (and the default route pointing to 2.2.2.254 will be put there), so the reverse traffic will go through the same ISP2.

Creating a vrouter and putting a second uplink in it is easy:

```
ISP2 {
    instance-type virtual-router;
    interface ge-0/0/2.0;
    routing-options {
        static {
            route 0.0.0.0/0 next-hop 2.2.2.254;
        }
    }
}
```

After doing it, however, DNAT for telnet sessions coming to 2.2.2.2 is still not working. Look at the routing table of the vrouter and you'll see that it does not have a route to internal subnet (192.168.50.0/24), which is clearly the reason for a failure:

```
lab@SRX> show route table ISP2

ISP2.inet.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[Static/5] 00:04:55
                    > to 2.2.2.254 via ge-0/0/2.0
2.2.2.0/24         *[Direct/0] 00:05:39
                    > via ge-0/0/2.0
2.2.2.2/32         *[Local/0] 00:05:39
                      Local via ge-0/0/2.0
```

So, somehow traffic needs to be directed from this instance to the IP of internal host (192.168.50.50) in the master routing instance. There are several ways to do that.

NOTE Starting from this point, you need to follow just one of the ways to send traffic to the internal host. The flexibility of Junos enables several methods of doing this, so if you are following the examples in your lab, save your configuration at this time, and start from this saved config in every sub-section that follows.

Method 1: Using RIB Groups

There currently is a route to 192.168.50.0/24 network only in the master routing instance (which uses routing table `inet.0`):

```
lab@SRX# run show route 192.168.50.0/24 exact

inet.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.50.0/24    *[Direct/0] 2d 06:20:18
                    > via ge-0/0/12.0
```

Let's copy it to vrouter's routing table, `ISP2.inet.0`:

```
[edit]
lab@SRX# set routing-options rib-groups CopyRoutes import-rib [inet.0 ISP2.inet.0]

[edit]
lab@SRX# set routing-options interface-routes rib-group inet CopyRoutes

[edit]
lab@SRX# set routing-instances ISP2 routing-options interface-routes rib-
group inet CopyRoutes

lab@SRX# commit
commit complete
```

You just created a RIB group (a set of two routing tables, in our case) and applied it to the interface routes of both main RIB (inet.0) and vrouter's RIB (ISP2.inet.0). Thus, you copied interface routes between tables (in both directions), which allows the traffic to flow easily between master instance and vrouter. In particular, the direct route to the internal network is now in both routing tables:

```
[edit]
lab@SRX# run show route 192.168.50/24 exact

inet.0: 9 destinations, 10 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.50.0/24    *[Direct/0] 2d 09:02:10
                    > via ge-0/0/12.0

ISP2.inet.0: 9 destinations, 9 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
192.168.50.0/24    *[Direct/0] 00:00:07
                    > via ge-0/0/12.0
```

Also, it is advantageous that the default route in the main instance shows two next hops in spite of the fact that the second uplink is now in a vrouter (this is not necessary for DNAT to work, however):

```
lab@SRX# run show route 0/0 exact

inet.0: 9 destinations, 10 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[Static/5] 2d 06:29:16
                    > to 1.1.1.254 via ge-0/0/1.0
                    [Static/7] 00:01:11
                    > to 2.2.2.254 via ge-0/0/2.0

ISP2.inet.0: 9 destinations, 9 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[Static/5] 02:53:43
                    > to 2.2.2.254 via ge-0/0/2.0
```

To our satisfaction, telnet from the REMOTE-HOST now works via both ISPs:

```
lab@REMOTE-HOST> telnet 1.1.1.1
Trying 1.1.1.1...
Connected to 1.1.1.1.
Escape character is '^]'.

INT-HOST (ttyp1)

login: ^C Client aborted login
Connection closed by foreign host.

lab@REMOTE-HOST> telnet 2.2.2.2
Trying 2.2.2.2...
Connected to 2.2.2.2.
Escape character is '^]'.

INT-HOST (ttyp1)

login: ^C Client aborted login
Connection closed by foreign host.
```

Method 2: Using Instance-import

You can achieve the same effects of copying routes between instances with the `instance-import` knob. First, write a policy that only accepts 192.168.50/24 route from master instance:

```
[edit]
lab@SRX# set policy-options policy-statement copy-int-
net term copy from instance master

[edit]
lab@SRX# set policy-options policy-statement copy-int-net term copy from route-
filter 192.168.50/24 exact accept
[edit]
lab@SRX# set policy-options policy-statement copy-int-net term else then reject
```

Now, apply the created policy to ISP2 instance:

```
[edit]
lab@SRX# set routing-instances ISP2 routing-options instance-import copy-int-net
```

After commit, check that the route is copied correctly:

```
[edit]
lab@SRX# run show route table ISP2

ISP2.inet.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[Static/5] 03:04:09
                    > to 2.2.2.254 via ge-0/0/2.0
2.2.2.0/24         *[Direct/0] 03:04:53
                    > via ge-0/0/2.0
2.2.2.2/32         *[Local/0] 03:04:53
                      Local via ge-0/0/2.0
192.168.50.0/24    *[Direct/0] 00:02:17
                    > via ge-0/0/12.0
```

Telnetting to both 1.1.1.1 and 2.2.2.2 from the remote host works as well (to save space, the corresponding command outputs are not shown here).

Method 3: Using Next-table

Let's drop the concept of copying routes between instances and just direct the traffic coming to the internal subnet of the master instance with the `next-table` option:

```
lab@SRX# set routing-instances ISP2 routing-options static route 192.168.50/24 next-
table inet.0
```

And DNAT starts working. Let's check with telnet, as usual, and the session table confirms:

```
lab@SRX# run show security flow session destination-prefix 2.2.2.2
Session ID: 41563, Policy name: DNAT2/5, Timeout: 1780, Valid
  In: 7.7.7.7/51162 --> 2.2.2.2/23;tcp, If: ge-0/0/2.0, Pkts: 10, Bytes: 671
  Out: 192.168.50.50/23 --> 7.7.7.7/51162;tcp, If: ge-0/0/12.0, Pkts: 9, Bytes: 643
Total sessions: 1
```

You can see from the output that DNAT is working because the destination address in the initiating flow of the session (2.2.2.2) is different from the source address for the return flow (192.168.50.50). Also you can see packets going in both directions, so everything is working just fine.

Method 4: Using Filter-based Forwarding

Yet another way to achieve the same result is to put a filter on the ge-0/0/2.0 interface that directs traffic to the main instance from the ISP2 vrouter:

```
lab@SRX# show firewall
family inet {
    filter to-main {
        term 10 {
            from {
                destination-address {
                    2.2.2.2/32;
                }
                destination-port 23;
            }
            then {
                routing-instance default; ## 'default' is not defined
            }
        }
        term 20 {
            then accept;
        }
    }
}

[edit]
lab@SRX# show interfaces ge-0/0/2
unit 0 {
    family inet {
        filter {
            input to-main;
        }
        address 2.2.2.2/24;
    }
}
```

As in previous cases, you will now see telnet from 7.7.7.7 to 2.2.2.2 as well as to 1.1.1.1 working fine.

NOTE You can ignore the warning in the filter saying that 'default' routing instance is not defined. Such a filter is actually directing the traffic to the master instance (tested with Junos 12.1X44-D10.4). If you are not comfortable with this warning, create another routing instance and copy routes from the master instance to it, then use the newly created instance in a firewall filter.

Method 5: Specifying Routing Instance in DNAT Pool

As a last method, you can just specify the routing instance in the DNAT pool:

```
 [edit]
lab@SRX# set security nat destination pool INT-HOST routing-instance default
```

This setting also directs traffic entering via ISP2 to the master instance:

```
lab@REMOTE-HOST> telnet 1.1.1.1
Trying 1.1.1.1...
Connected to 1.1.1.1.
Escape character is '^]'.

INT-HOST (ttyp1)

login: ^C Client aborted login
Connection closed by foreign host.

lab@REMOTE-HOST> telnet 2.2.2.2
Trying 2.2.2.2...
Connected to 2.2.2.2.
Escape character is '^]'.

INT-HOST (ttyp1)

login: ^C Client aborted login
Connection closed by foreign host.
```

This last option turns out to be another short way to achieve the task.

MORE? There is at least one more way to achieve the task of connecting ISP2 vrouter with the main instance: by using logical tunnel (LT) interfaces that Junos provides. Search the Junos documentation if you want more details.

Summary

Setting up destination NAT in a dual ISP environment can be accomplished via a number of different options. The particular option you choose depends on your preferences, as well as other setup requirements; you will, however, need to implement routing instances to solve the problem with reverse route lookup.

It is important to remember that the SRX Series, being a stateful device, cares about each traffic session. Routing lookup is performed only when a first packet of the session arrives, and both the route for the traffic destination as well as the reverse route for the traffic source are needed to create a session. Once the session is created, the packets for all subsequent sessions are forwarded accordingly.

Recipe 13

Rapid Port Templating

Problem

Configuring multiple ports on a device such as the EX Series can be tedious and changes are vulnerable to human error.

Solution

One of the nice features of Junos is the ability to build template configurations very easily and then deploy them rapidly across the device.

A great example of this is the interface-range statement provided on the EX platform. By grouping ports with identical roles as members of an interface range, you can change a piece of configuration within the interface range and have it instantly applied to all member interfaces. For instance:

```
interfaces {
    interface-range MY-RANGE {
        member ge-0/0/13;
        member ge-0/0/14;
        member ge-0/0/15;
        member ge-0/0/16;
        member ge-0/0/17;
        member ge-0/0/18;
        unit 0 {
            family ethernet-switching {
                vlan {
                    members v100-Workstations;
                }
            }
        }
    }
}
{master:0}[edit]
bdale@ex42-lab01# run show vlans v100-Workstations
Name           Tag     Interfaces
v100-Workstations 100
                       ge-0/0/13.0, ge-0/0/14.0, ge-0/0/15.0,
                       ge-0/0/16.0, ge-0/0/17.0, ge-0/0/18.0
```

Similarly, when configuring protocols, you can reference an interface range and have the protocol configuration applied to every member automatically, as shown here:

```
{master:0}[edit]
bdale@ex42-lab01# show ethernet-switching-options
voip {
    interface MY-RANGE {
        vlan v101-VoIP;
        forwarding-class expedited-forwarding;
    }
}
{master:0}[edit]
bdale@ex42-lab01# show protocols rstp
bridge-priority 8k;
interface MY-RANGE {
    edge;
}
bpdu-block-on-edge;
```

This may seem trivial at first, but imagine you have four fully-populated EX8216 chassis configured as a single virtual chassis – you could have over 3,000 physical interfaces in a single box!

Now, when configuring your `interface-range` statements, it may be tempting to use the `member-range` statement to limit the amount of configuration you need to enter:

```
interface-range SERVER-PORTS {
    member-range ge-0/0/8 to ge-0/0/20;
    member-range ge-1/0/8 to ge-1/0/20;
    member-range ge-2/0/8 to ge-2/0/20;
    member-range ge-3/0/8 to ge-3/0/20;
    member-range ge-4/0/8 to ge-4/0/20;
    member-range ge-5/0/8 to ge-5/0/20;
    unit 0 {
        family ethernet-switching {
            vlan {
                members v200-Infrastructure;
            }
        }
    }
}
```

However, if you need to make a change later, things get messy pretty quickly. For instance, let's say that port ge-3/0/13 is no longer a server port:

```
{master:0}[edit]
bdale@ex42-lab01# delete interfaces interface-range SERVER-PORTS member-range ge-
3/0/8 to ge-3/0/20
{master:0}[edit]
bdale@ex42-lab01# set interfaces interface-range SERVER-PORTS member-range ge-
3/0/8 to ge-3/0/12
{master:0}[edit]
bdale@ex42-lab01# set interfaces interface-range SERVER-PORTS member-range ge-
3/0/14 to ge-3/0/20
{master:0}[edit]
bdale@ex42-lab01# show interfaces | find SERVER-PORTS
interface-range SERVER-PORTS {
    member-range ge-0/0/8 to ge-0/0/20;
    member-range ge-1/0/8 to ge-1/0/20;
    member-range ge-2/0/8 to ge-2/0/20;
    member-range ge-4/0/8 to ge-4/0/20;
    member-range ge-5/0/8 to ge-5/0/20;
    member-range ge-3/0/8 to ge-3/0/12;
    member-range ge-3/0/14 to ge-3/0/20;
    unit 0 {
        family ethernet-switching {
            vlan {
                members v200-Infrastructure;
            }
        }
    }
}
```

Now, let's say you need to remove ge-3/0/15 as well, as it is no longer a server port either:

```
{master:0}[edit]
bdale@ex42-lab01# delete interfaces interface-range SERVER-PORTS member-range ge-
3/0/14 to ge-3/0/20
{master:0}[edit]
bdale@ex42-lab01# set interfaces interface-range SERVER-PORTS member ge-3/0/14
{master:0}[edit]
bdale@ex42-lab01# set interfaces interface-range SERVER-PORTS member-range ge-
3/0/16 to ge-3/0/20
{master:0}[edit]
bdale@ex42-lab01# show interfaces | find SERVER-PORTS
interface-range SERVER-PORTS {
    member ge-3/0/14;
    member-range ge-0/0/8 to ge-0/0/20;
    member-range ge-1/0/8 to ge-1/0/20;
    member-range ge-2/0/8 to ge-2/0/20;
    member-range ge-4/0/8 to ge-4/0/20;
    member-range ge-5/0/8 to ge-5/0/20;
    member-range ge-3/0/8 to ge-3/0/12;
    member-range ge-3/0/16 to ge-3/0/20;
    unit 0 {
        family ethernet-switching {
            vlan {
                members v200-Infrastructure;
            }
        }
    }
}
```

As you can see, this gets messy very quickly and is just prime for an operator error! Fortunately, there is an easier way, by creating your interface ranges using only the `member` option, but saving all the typing by using the `wildcard range set` command (available from Junos 12.1):

```
{master:0}[edit]
bdale@ex42-lab01# wildcard range set interfaces interface-range SERVER-
PORTS member ge-[0-5]/0/[8-20]
|
{master:0}[edit]
bdale@ex42-lab01# show interfaces | find SERVER-PORTS
interface-range SERVER-PORTS {
    member ge-0/0/8;
    member ge-0/0/9;
    member ge-0/0/10;
    member ge-0/0/11;
    member ge-0/0/12;
    member ge-0/0/13;
    member ge-0/0/14;
...
    member ge-5/0/14;
    member ge-5/0/15;
    member ge-5/0/16;
    member ge-5/0/17;
    member ge-5/0/18;
    member ge-5/0/19;
    member ge-5/0/20;
    unit 0 {
        family ethernet-switching {
            vlan {
                members v200-Infrastructure;
            }
        }
    }

}
```

Now you can simply add and delete individual members from your interface range with a single command.

TIP You can also use `wildcard range delete` if you have a large number of consecutive ports to remove via a regular expression.

Finally, if you need to disable a specific protocol configuration on one or more ports, but you need a logical configuration such as VLANs to be maintained, you can override individual ports per protocol. For example:

```
{master:0}[edit]
bdale@ex42-lab01# show protocols rstp
bridge-priority 8k;
interface ge-0/0/14.0 {
    disable;
}
interface SERVER-PORTS {
    edge;
}
bpdu-block-on-edge;

{master:0}[edit]
bdale@ex42-lab01# run show spanning-tree interface

Spanning Tree interface parameters for instance 0

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID        bridge ID          Cost
...
ge-0/0/13.0    128:526      128:526   8192.80711fe21701     20000  FWD    DESG
ge-0/0/14.0    128:527      128:527  32768.80711fe21701     20000  DIS    DIS
```

Recipe 14

SRX to ASA Policy Based IPSec VPN

Problem

Keeping an organization's network and data safe is paramount, and firewalls can help, but they also introduce a new issue: How does legitimate traffic get in? The simple answer is to use a site-to-site VPN, but the configuration requires a number of considerations that are by no means simplistic. Let's investigate.

Firewalls are designed to keep unwanted traffic out of your network, but also to allow users inside the network to dynamically access resources outside the network. What happens however, if a user inside your network needs to access several resources inside another network?

Solution

This is where site-to-site VPNs are used. These allow secure encrypted 'tunnels' across the Internet, but it was recognized early on that there are many manufacturers of firewalls, and connecting them requires a common set of encryption protocols. Thankfully the IETF has already thought of this, and today the most common VPN in use utilizes the *IPSec protocol suite*.

MORE? This recipe details how to connect devices via IPSec VPN. It is not a study guide on IPSec itself, nor does it cover IPSec in any great detail. Should you need to learn more about IPSec, search on *IPSec VPN Junos* at www.juniper.net/techpubs.

Here is this recipe's topology. One site is using a Juniper SRX100 with a public facing IP address of 150.34.18.34, and behind the firewall is a private subnet using the network address of 10.100.100.0/24. The second site will be using a Cisco ASA5505 with an IP address of 3.18.215.10 and behind the firewall will be a private subnet using the network address of 172.23.7.0/24.

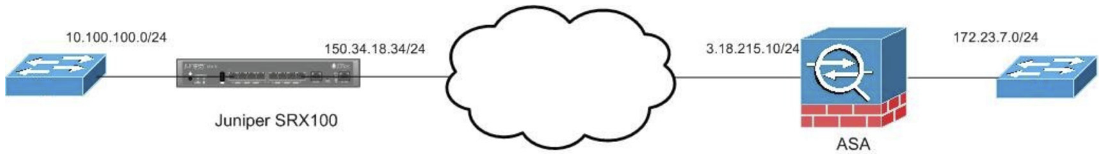


Figure 14.1 This Recipe's Topology

The configurations of these devices are as follows:

SRX

```
fe-0/0/0 {
    unit 0 {
        family inet {
            address 150.34.18.34/24;
        }
    }
}
fe-0/0/2 {
    unit 0 {
        family inet {
            address 10.100.100.1/24;
        }
    }
}
```

ASA

```
interface Ethernet0/0
 switchport access vlan 300
!
interface Ethernet0/2
 switchport access vlan 200
!
interface Vlan200
 nameif inside
 security-level 100
 ip address 172.23.7.1 255.255.255.0
!
interface Vlan300
 nameif outside
 security-level 0
 ip address 3.18.215.10 255.255.255.0
```

In addition, the SRX also has the following configuration to specify which interface is in the ‘trusted’ zone and which is in the ‘untrusted’ zone:

```
 [edit security zones]
root@SRX100# show
security-zone trust {
   host-inbound-traffic {
      system-services {
         all;
      }
      protocols {
         all;
      }
   }
   interfaces {
      vlan.0;
      fe-0/0/2.0;
   }
}
security-zone untrust {
   screen untrust-screen;
   interfaces {
      fe-0/0/0.0 {
         host-inbound-traffic {
            system-services {
               dhcp;
               tftp;
```

NOTE In this scenario, both firewalls have a default route pointing to their respective ISPs as their next hop.

Discussion

The goal of this recipe is to allow users in 10.100.100.0/24 to access resources in 172.23.7.0/24 and vice versa, securely, with a guarantee that the data has not been tampered with. So, the first step would be to ensure the devices can reach each other.

Let’s ping the ASA from the SRX:

```
root@SRX100> ping 3.18.215.10
PING 3.18.215.10 (3.18.215.10): 56 data bytes
64 bytes from 3.18.215.10: icmp_seq=0 ttl=254 time=2.438 ms
64 bytes from 3.18.215.10: icmp_seq=1 ttl=254 time=2.245 ms
64 bytes from 3.18.215.10: icmp_seq=2 ttl=254 time=2.395 ms
```

If the reverse is attempted, there is no response because the SRX disallows pings. Therefore, just for testing purposes:

```
set security zones security-zone untrust interfaces fe-
0/0/0.0host-inbound-traffic system-services ping
```

```
Commit
```

CAUTION By default the SRX will not respond to pings so that an attacker cannot easily find its IP address and attack it. Allowing a ping could compromise this security feature, therefore, please do it with caution or as with most of the recipes in this book, they should be tried on nonproduction environments.

Now ping the SRX from the ASA and you'll get:

```
ASA# ping 150.34.18.34
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.34.18.34, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

Now, let's remove the ping option as a best practice:

```
Delete security zones security-zone untrust interfaces fe-0/0/0.0 host-inbound-traffic
system-services ping
Commit
```

To be certain there is network reachability, the next step would be to decide which policies to use for IPSec, and it's an important step because the encryption needs to be strong, but supported by both devices. In addition, the VPN connection is done in two phases: Phase 1 is called *IKE*, which initializes the connection and performs a secure method with which to exchange the encryption keys, and Phase 2, which is known as *IPSec* and which performs the actual encryption of the data sent between the sites.

With IKE version 1, there are four different values for the firewall to use when establishing the connection. These relate to encryption, authentication, the Diffie-Hellman Group used to calculate a shared secret, and data integrity, and some of the options for these are listed here:

Value	Possible Options
Diffie-Hellman Group	1 2 5 7
Authentication	Pre-Shared RSA Signature Crack

Encryption	DES 3DES AES – 128bit AES – 192bit AES – 256bit
Data Integrity	MD5 SHA

And IPSec has three values you need to supply: protocol, encryption, and Data Integrity, as listed here:

Value	Possible Options
Data Integrity	SHA1 SHA256 SHA384 SHA512 MD5
Encryption Algorithm	DES 3DES AES – 128bit AES – 192bit AES – 256bit
Protocol	ESP AH

NOTE Not all these values are supported on every device. It is up to the individual engineer to decide which options should be used and which cannot be used before beginning the configuration.

This recipe uses Diffie-Hellman group 2, Pre-Shared Key, 3DES, and SHA for the IKE Policy. This is created on the SRX by issuing the following commands:

```
set security ike proposal SITE2SITEIKE authentication-method pre-shared-keys
set security ike proposal SITE2SITEIKE encryption-algorithm 3des-cbc
set security ike proposal SITE2SITEIKE dh-group group2
set security ike proposal SITE2SITEIKE authentication-algorithm sha1
```

This policy needs to be mirrored on the ASA.

```
crypto ikev1 policy 10
authentication pre-share
encryption 3des
hash sha
group 2
```

For the IPSec proposal, let's use ESP, SHA1, and 3DES, entered into the SRX:

```
set security ipsec proposal SITE2SITEPROP protocol esp
set security ipsec proposal SITE2SITEPROP authentication-algorithm hmac-sha1-96
set security ipsec proposal SITE2SITEPROP encryption-algorithm 3des-cbc
```

Again, it is mirrored on the ASA so that the proposals match:

```
crypto ipsec ikev1 transform-set SITE2SITEPROP esp-3des esp-sha-hmac
```

Part of creating a VPN is to discover which traffic should be sent across the encrypted tunnel and which should not. For example, if a user was accessing a website, it is a waste of bandwidth to send the request across the encrypted tunnel then out onto the Internet. This would cause an unnecessary delay, therefore you need to tell the firewalls which traffic to encrypt, also known as classifying the 'interesting' traffic.

On the SRX, create what is known as an *address book*, which is just giving the interesting traffic a name, an easier reference later than a list of subnets:

```
set security zones security-zone trust address-book address SRXNETWORK 10.100.100.0/24
set security zones security-zone untrust address-book address ASANETWORK 172.23.7.0/24
```

On the ASA you would use an ACL:

```
access-list SITE2SITETRAFFIC line 1 extended permit ip 172.23.7.0 255.255.255.0
10.100.100.0 255.255.255.0
```

Once these policies and proposals have been created, and the interesting traffic has been classified, this information can then be joined in a policy or map to tell the firewall to use these policies and proposals. In addition, the firewall needs to be told the IP address of the peer and what the pre-shared key is.

The SRX and ASA do this slightly differently. In the SRX, a security policy would be created to tell the SRX to send the traffic destined for the remote site through the tunnel and which tunnel to use. The issue here is that by default, the SRX has a security policy in place to permit all outgoing traffic. As this policy is first in the list, our new policy will be ignored. This can be corrected by using the command `insert policy` as shown at the end of this configuration:

```
set security policies from-zone trust to-zone untrust policy VPNPOLICY match source-
address SRXNETWORK
set security policies from-zone trust to-zone untrust policy VPNPOLICY match
destination-address ASANETWORK
set security policies from-zone trust to-zone untrust policy VPNPOLICY match
application any
set security policies from-zone trust to-zone untrust policy VPNPOLICY then permit
```

```
tunnel ipsec-vpn SITE2SITEVPN
set security policies from-zone trust to-zone untrust policy VPNPOLICY then permit
tunnel pair-policy POLICYVPNREVERSE
insert policy VPNPOLICY before policy trust-to-untrust
```

In addition, it is necessary to tell the SRX how to treat the return traffic coming from the ASA, as follows:

```
set security policies from-zone untrust to-zone trust policy POLICYVPNIN match source-
address ASANETWORK
set security policies from-zone untrust to-zone trust policy POLICYVPNIN match
destination-address SRXNETWORK
set security policies from-zone untrust to-zone trust policy POLICYVPNIN match
application any
set security policies from-zone untrust to-zone trust policy POLICYVPNIN then permit
tunnel ipsec-vpn SITE2SITEVPN
set security policies from-zone untrust to-zone trust policy POLICYVPNIN then permit
tunnel pair-policy VPNPOLICY
```

For the ASA, however a crypto map is used:

```
crypto map outside_map 10 match address SITE2SITETRAFFIC
```

As the policies and profiles we created earlier are bound together, it soon becomes obvious that it is a good idea to give the policies and profiles names that allow easy identification as to what their purpose is. It would have been very simple to just call every policy "SI-TE2SITE", but, as is obvious in the next step of configuring the VPN that is creating the IKE policy, the second command would have read "set security ike policy SITE2SITE proposals SITE2SITE". This may cause confusion later, especially from a support perspective.

```
set security ike policy SITE2SITEPOL mode main
set security ike policy SITE2SITEPOL proposals SITE2SITEIKE
set security ike policy SITE2SITEPOL pre-shared-key ascii-text SUPERSECRET
```

The ASA uses a *group policy* as opposed to an IKE policy. Under this policy, the VPN protocol that the firewall should use is entered. In this case, *ikev1* is used as opposed to IKE version 2, which is more related to "AnyConnect" VPNs, whereas IPSec uses IKE version 1.

```
group-policy SITE2SITEPOLICY internal
group-policy SITE2SITEPOLICY attributes
vpn-tunnel-protocol ikev1
```

For a VPN device to connect to another VPN device, it needs to be told the address of the device it is to connect to, known as its *peer*. The device should also be told from which interface it should connect to its peer. This step may seem pointless at first, but you can specify that a loopback should be used instead of an outside interface. On the SRX, an *IKE Gateway* is created for this purpose:

NOTE An IPSec policy has been included under the Gateway configuration. It's included here as the ASA adds its IPSec policy under the crypto map.

```
set security ike gateway SITE2SITEGW ike-policy SITE2SITEPOL
set security ike gateway SITE2SITEGW address 3.18.215.10
set security ike gateway SITE2SITEGW external-interface fe-0/0/0
set security ipsec policy SITE2SITEIPSECPOL proposals SITE2SITEPROP
```

The ASA uses a crypto map to perform the same function:

```
crypto map outside_map 10 set peer 150.34.18.34
crypto map outside_map 10 set ikev1 transform-set SITE2SITEPROPOSAL
crypto map outside_map interface outside
```

The next step is to create a VPN telling the SRX which Gateway should be used and which IPSec policy to use. Let's also tell the SRX to create the tunnel now by adding the "establish-tunnels immediately" option. The alternative is to use the "establish-tunnels on-traffic" option whereby the tunnel is not created until the SRX receives traffic that needs to traverse the VPN:

```
set security ipsec vpn SITE2SITEVPN ike gateway SITE2SITEGW
set security ipsec vpn SITE2SITEVPN ike ipsec-policy SITE2SITEIPSECPOL
set security ipsec vpn SITE2SITEVPN establish-tunnels immediately
```

A tunnel group is created on the ASA for the same purpose. Note that the pre-shared key is placed under the tunnel group as opposed to under the IKE policy as it is on the SRX:

```
tunnel-group 150.34.18.34 type ipsec-l2l
tunnel-group 150.34.18.34 general-attributes
 default-group-policy SITE2SITEPOLICY
tunnel-group 150.34.18.34 ipsec-attributes
 ikev1 pre-shared-key SUPERSECRET
```

One final thing that needs to be done is that the firewall needs to be told to allow VPN connections in on its outside interface. The system service "ike" is added to the untrusted interface in the security zone on the SRX:

```
set security zones security-zone untrust interfaces fe-0/0/0.0 host-inbound-traffic
system-services ike
```

ASA 5505s use *nameif* interfaces, and in this case, the nameif is called *outside*. The ASA should be instructed to enable IKE version 1 on this interface:

```
crypto ikev1 enable outside
```

Now all necessary configurations are complete, a simple check can be made to see whether the VPN tunnel has been created. To do this, the show security ike security-associations command and show security ipsec security-associations command are used:

```
root@SRX100> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode    Remote Address
6427052 UP     d85a77ee7a776645  8e2e7c546c6140c6  Main    3.18.215.10

root@SRX100> show security ipsec security-associations
  Total active tunnels: 1
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <4    ESP:3des/sha1   32795eb2 2672/ unlim   -   root 500   3.18.215.10
  >4    ESP:3des/sha1   2787d214 2672/ unlim   -   root 500   3.18.215.10
```

You can see that the session appears to be up.

By entering the show security ike security-association detail and show security ipsec security-association detail commands you can see IKE traffic statistics and check that the correct interesting traffic has been classified:

```
root@SRX100> show security ike security-associations detail
IKE peer 3.18.215.10, Index 6427053, Gateway Name: SITE2SITEGW
  Role: Initiator, State: UP
  Initiator cookie: d27209a4f394e9b9, Responder cookie: d85ec42d67a83ea5
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 150.34.18.34:500, Remote: 3.18.215.10:500
  Lifetime: Expires in 28603 seconds
  Peer ike-id: 3.18.215.10
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : 3des-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-2
  Traffic statistics:
   Input  bytes  :                2264
   Output bytes  :                2408
   Input  packets:                  23
   Output packets:                  24
  Flags: IKE SA is created
  IPSec security associations: 1 created, 0 deleted
  Phase 2 negotiations in progress: 0

    Negotiation type: Quick mode, Role: Initiator, Message ID: 0
    Local: 150.34.18.34:500, Remote: 3.18.215.10:500
    Local identity: 150.34.18.34
    Remote identity: 3.18.215.10
    Flags: IKE SA is created
```

```
root@SRX100> show security ipsec security-associations detail
  ID: 4 Virtual-system: root, VPN Name: SITE2SITEVPN
  Local Gateway: 150.34.18.34, Remote Gateway: 3.18.215.10
  Local Identity: ipv4_subnet(any:0,[0..7]=10.100.100.0/24)
  Remote Identity: ipv4_subnet(any:0,[0..7]=172.23.7.0/24)
  Version: IKEv1
    DF-bit: clear
    Policy-name: VPNPOLICY
  Port: 500, Nego#: 7, Fail#: 0, Def-Del#: 0 Flag: 600829
  Tunnel Down Reason: Cleared via CLI
    Direction: inbound, SPI: 6f91abf3, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 3583 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2945 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64

    Direction: outbound, SPI: 3d7dd89c, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 3583 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2945 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
```

Although the security associations are up, there is one small issue: hosts in the remote site cannot access resources local to the opposing site. The reason for this is simple. Since these firewalls are connected to the Internet and the subnet behind the firewall is using an RFC1918 address, the firewalls are configured to use NAT. Unfortunately, this means that the firewalls will also attempt to use NAT when sending traffic across the VPN. The firewalls need to be notified that when sending traffic from 10.100.100.0/24 to 172.23.7.0/24, or from 172.23.7.0/24 to 10.100.100.0/24, not to use NAT.

By default the SRX already has NAT configured. A new rule needs to be created to specify the source and destination, then the insert command should be used to place the new rule ahead of the default rule;

```
set security nat source rule-set trust-to-untrust rule VPN match source-address 
10.100.100.0/24 destination-address 172.23.7.0/24
set security nat source rule-set trust-to-untrust rule VPN then source-nat off
insert security nat source rule-set trust-to-untrust rule VPN before rule source-nat-
rule
```

Back on the ASA, NAT is not enabled by default, therefore "objects" should be created to classify the subnets, then the inside subnet is added to a NAT statement so that normal internet traffic still uses NAT, then the exceptions are created by the use of a "Double NAT" statement:

```
object network SRXHOST
 subnet 10.100.100.0 255.255.255.0
object network INSIDENAT
 subnet 172.23.7.0 255.255.255.0
 nat (inside,outside) dynamic interface
nat (outside,inside) source static SRXHOST SRXHOST destination static INSIDENAT
INSIDENAT
nat (inside,outside) source static INSIDENAT INSIDENAT destination static SRXHOST
SRXHOST
```

NOTE The ASA used in this recipe was using version 8.4. NAT in 8.4 is completely different from ASA version 8.2, therefore if you are unsure as to the reason why a *double NAT* is performed, it is strongly recommended you consult the relevant Cisco documentation.

Now that NAT has been configured, it is possible to send encrypted traffic between the subnets. Run the `show security ipsec statistics` command several times and it should show the encrypted bytes increasing, thus proving the VPN is now working normally:

```
root@SRX100> show security ipsec statistics
ESP Statistics:
  Encrypted bytes:          6373848
  Decrypted bytes:           140575
  Encrypted packets:          38056
  Decrypted packets:           1997
AH Statistics:
  Input bytes:                    0
  Output bytes:                   0
  Input packets:                  0
  Output packets:                 0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0
```

Recipe 15

Hub and Spoke VPN with OSPF

Problem

Hub and spoke VPNs provide an effective way to provide central office or data center resources to remote offices over the Internet. Each site connects to the central service location using IPSEC VPN and can then access these published services.

But standard policy-based VPNs, or those using static routing, present additional management overhead, and full mesh routing requires that new routes be created at each site for any new spoke added to the network.

Solution

To solve this predicament, route-based VPNs are used along with OSPF to automatically create and distribute routes for each site.

The configuration's basic steps are:

- Configure the hub for needed services
- Configure multi-point tunnel interface
- Configure hub LAN interface
- Enable OSPF settings

For each spoke:

- Configure the hub VPN parameters for the spoke
- Configure the spoke LAN interface
- Configure the spoke VPN parameters
- Configure OSPF on Spoke

NOTE This recipe was tested on a SRX100 using Junos 11.4r5.5 and that all configuration statements are shown as they would be entered on the branch device.

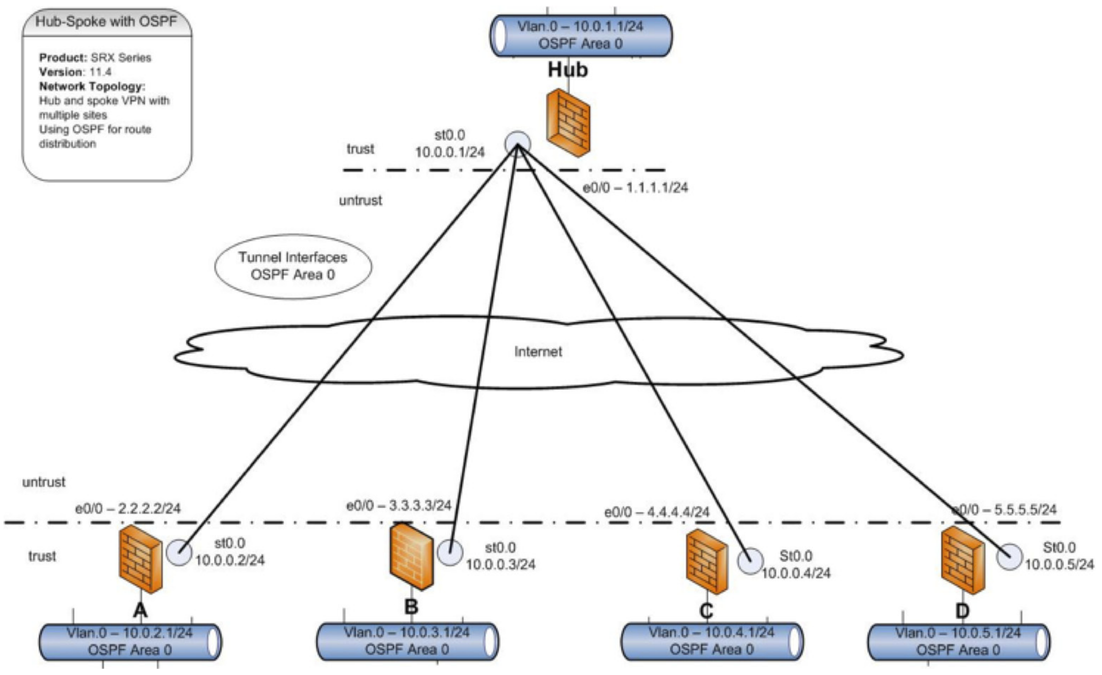


Figure 15.1 This Recipe's Topology

Configure the hub multi-point tunnel interface

Create the VPN tunnel interface as multipoint and assign an IP range and address:

```
set interfaces st0 unit 0 family inet address  10.0.0.1/24
set interfaces st0 unit 0 multipoint
set zones security-zone trust interfaces st0.0
```

Configure the hub LAN interface

Create a LAN vlan, place it in the trust zone, and assign the desired interfaces:

```
set interfaces fe-0/0/1 unit 0 family ethernet-switching
set interfaces fe-0/0/1 unit 0 vlan members vlan-trust
set interfaces vlan unit 0 family inet address 10.0.1.1/24;
set vlans vlan-trust vlan-id 0
set vlans vlan-trust l3-interface vlan.0
set zones security-zone trust host-inbound-traffic system-services all
set zones security-zone trust interfaces vlan.0
```

Configure the hub OSPF parameters

Enable OSPF on the hub SRX and assign the local vlan interface and the tunnel interface to OSPF area 0:

```
set protocols ospf area 0 interface vlan.0
set protocols ospf area 0 interface st0.0
```

Allow the host services for OSPF on the interfaces:

```
set security zones security-zone trust interfaces vlan.0 host-inbound-traffic
protocols ospf
set security zones security-zone trust interfaces st0.0 host-inbound-traffic protocols
ospf
```

Configure vlan.0 to announce OSPF routes:

```
set protocols ospf area 0 interface vlan.0 passive
```

Configure the OSPF router-id for the hub:

```
set routing-options router-id 10.0.1.1
```

Configure the hub VPN parameters for the spoke

Start by configuring the local gateway interface for the VPN. This will be the public IP address and interface from the local Internet service, and place these into the untrust zone:

Configure the Internet static IP address on the gateway interface:

```
set interfaces fe-0/0/0 unit 0 family inet address 1.1.1.1/24
```

Now set the default route for the local Internet service:

```
set routing-options static route 0.0.0.0/0 next-hop 1.1.1.254
```

Place the gateway interface into the untrust zone:

```
set security zones security-zone untrust interfaces fe-0/0/0.0
```

Now allow management services on this interface for the VPN IKE connections. You can also allow management services as needed on this interface, but the IKE services are required to establish the VPN connection. Since this is an Internet-facing interface, you will not enable any other system services for now:

```
set security zones security-zone untrust host-inbound-traffic system-services ike
```

Next, configure the remote gateway IKE parameters for policy and a preshared key. These can be used by all the spoke VPN connections and are created on the hub just once:

```
set security ike policy ike-policy1 mode main
set security ike policy ike-policy1 proposal-set standard
set security ike policy ike-policy1 pre-shared-key ascii-text "sharedsecret"
```

Now create the IPSEC parameters for the VPN connection. Again, these are used by all the spoke connections and are created just once:

```
set security ipsec policy vpn-policy1 proposal-set standard
```

Follow these steps to create the route-based VPN using the tunnel interface you just created. The steps need to be repeated for each spoke site added to the system, but for each spoke site the IP address of the remote gateway will need to be adjusted.

```
set security ike gateway ike-gate ike-policy ike-policy1
```

For this remote IKE gateway address, the IP address will change for each new spoke added.

```
set security ike gateway ike-gate address 2.2.2.2
set security ike gateway ike-gate external-interface fe-0/0/0.0
```

Then use the existing policy to create the IPSEC Phase 2 connection and tie this to the tunnel interface:

```
set security ipsec vpn ike-vpn ike gateway ike-gate
set security ipsec vpn ike-vpn ike ipsec-policy vpn-policy1
set security ipsec vpn ike-vpn bind-interface st0.0
```

Finally, set the VPN MTU to the most common value to prevent fragmentation issues of traffic across the tunnel:

```
set security flow tcp-mss ipsec-vpn mss 1350
```

Configure the Spoke LAN interface

Create the LAN vlan, place it in the trust zone, and assign the desired interfaces. You'll need to adjust the vlan IP address per each spoke:

```
set interfaces fe-0/0/1 unit 0 family ethernet-switching
set interfaces fe-0/0/1 unit 0 vlan members vlan-trust
set interfaces vlan unit 0 family inet address 10.0.2.1/24
set vlans vlan-trust vlan-id 0
set vlans vlan-trust l3-interface vlan.0
set zones security-zone trust host-inbound-traffic system-services all
set zones security-zone trust interfaces vlan.0
```

Configure the spoke VPN parameters

Next, create the tunnel interface for use in the VPN connection.

Adjust the IP address to match the particular spoke you are configuring per the diagram:

```
set interfaces st0 unit 0 family inet address  10.0.0.X/24
```

Follow these steps to create the route-based VPN using the tunnel interface created above. Start by configuring the local gateway interface for the VPN. This will be the public IP address and interface from the local Internet service and place these into the untrust zone.

Configure the Internet static IP address on the gateway interface, and this address changes per the spoke being configured:

```
set interfaces ge-0/0/0 unit 0 family inet address 2.2.2.2/24
```

Now, set the default route for our local Internet service:

```
set routing-options static route 0.0.0.0/0 next-hop 2.2.2.254
```

Place this gateway interface into the untrust zone:

```
set security zones security-zone untrust interfaces fe-0/0/0.0
```

Now let's allow management services on this interface for the VPN IKE connections. You can also allow management services as needed on this interface, but the IKE services are required to establish the VPN connection. But since this is an Internet-facing interface, let's not enable any other system services for now.

```
set security zones security-zone untrust host-inbound-traffic system-services ike
```

Next, configure the remote gateway IKE parameters for policy and the preshared key. Naturally, these parameters must match the configuration at the corporate site where the VPN is established.

```
set security ike policy ike-policy1 mode main
set security ike policy ike-policy1 proposal-set standard
set security ike policy ike-policy1 pre-shared-key ascii-text "sharedsecret"
```

Now we use this policy set to create the ike gateway:

```
set security ike gateway ike-gate ike-policy ike-policy1
```

This is the gateway of the hub so this will remain the same on all the spoke devices:

```
set security ike gateway ike-gate address 1.1.1.1
set security ike gateway ike-gate external-interface fe-0/0/0.0
```

Now create the IPSEC parameters for the VPN connection:

```
set security ipsec policy vpn-policy1 proposal-set standard
```

Now use this policy to create the IPSEC Phase 2 connection and tie this to the tunnel interface:

```
set security ipsec vpn ike-vpn ike gateway ike-gate
set security ipsec vpn ike-vpn ike ipsec-policy vpn-policy1
set security ipsec vpn ike-vpn bind-interface st0.0
```

Finally, set the VPN MTU to the most common value, preventing fragmentation of traffic across the tunnel:

```
set security flow tcp-mss ipsec-vpn mss 1350
```

Configure the Spoke OSPF Parameters

Follow these steps on all spoke devices. For the LAN IP addresses, substitute the spoke subnet for the X in the configuration statement.

Enable OSPF on the hub SRX and assign the local vlan interface and the tunnel interface to OSPF area 0:

```
set protocols ospf area 0 interface vlan.0
set protocols ospf area 0 interface st0.0
```

Allow the host services for OSPF on the interfaces:

```
set security zones security-zone trust interfaces vlan.0 host-inbound-traffic protocols
ospf
set security zones security-zone trust interfaces st0.0 host-inbound-traffic protocols
ospf
```

And finally, configure vlan.0 to announce OSPF routes:

```
set protocols ospf area 0 interface vlan.0 passive
configure the OSPF router-id for the hub.  Substitute the subnet for the particular hub.
set routing-options router-id 10.0.X.1
```

References

Understanding IPSEC VPN

http://www.juniper.net/techpubs/en_US/junos11.4/information-products/topic-collections/security/software-all/security/index.html?topic-54272.html

Configuring Route Based VPN

http://www.juniper.net/techpubs/en_US/junos11.4/information-products/topic-collections/security/software-all/security/index.html?topic-52842.html

Recipe 16

Three Things to Check on a SSL VPN

There are hundreds of calls to Juniper Support that are unnecessary if you first check a few things on your SSL VPN. Review these three operational fundamentals that often get overlooked in production environments.

Are You Running the Latest Code?

Juniper releases at least one major version of the SSL VPN IVE operating system, along with several minor releases, every year. Each version brings new features and routine fixes, but crucially it also includes support for newly released browsers and client operating systems. Inevitably users will update to the latest the software on workstations and tablets. Generally the client-side components of the Juniper MAG family are resilient to change but sometimes specific support is needed triggering a code upgrade on the IVE. In a well-maintained environment you can expect to upgrade the software two or three times a year. With a cluster deployment this upgrade can be performed without downtime.

By keeping the software up-to-date you can ensure that your users always get the best possible experience and your investment in SSL VPN is protected.

Have You Installed the Juniper Installer Service?

Microsoft Windows requires administrator privileges for some software installation changes and upgrades used by the various Juniper endpoint agents. Many organizations do not give laptop users local administrator privileges: to allow roaming users to remotely upgrade, Juniper provides the Juniper installer service. Once installed by an administrator or delivered as a part of remotely-deployed software upgrade, the Juniper client components will silently upgrade themselves even if the user is not a local administrator. The package is easy to install but works even better when combined with a white list file located in *%ProgramFiles%\Juniper Networks\Whitelist.txt* that contains a list of the public-facing URLS of your IVE appliances. For example:

```
Primary.juniper.net
Secondary.juniper.net
192.168.0.100
192.168.0.100
```

The Juniper installer service and white list file can improve your user experience when performing routine upgrades.

Is Your System Time Synchronized?

When troubleshooting authentication issues, it's important to understand when, and in what order, events are occurring. Log files and policy traces can only be meaningful when the date and time is consistent across the client, the IVE, and the server back-end. Some authentication protocols (such as RSA SecureID) just won't work if the time is not synchronized correctly. The easiest way to address this is to utilize the built in Network Time Protocol (NTP) client on the IVE and configure it to use a reliable time source. Many organizations utilize Active Directory domain controllers or even a core router as a time server, but whatever you use, make sure that it is consistent across the network. Setting up time service is a simple change that is often overlooked and can reduce the effort required for troubleshooting.