Junos® Dynamic Services Series

DAY ONE: DEPLOYING SRX SERIES SERVICES GATEWAY

It's day one and you need to deploy your new security device running Junos. Get it done today with this practical, time-saving book that shows you what you need to do.

By Barny Sanchez

DAY ONE:
DEPLOYING SRX SERIES SERVICES GATEWAY

Find out what Junos can do running a security device. The SRX devices are super-charged firewalls fortified with routing and switching capabilities. So get ready because working on the SRX Series combines powerful Junos networking with a potent set of new security services. This book shows you how to get started: how to console to your SRX device, perform initial configuration, and deploy your new box in a matter of hours. There's no theory, no workarounds, no chatty diversions. Let's get going.

"The SRX is an extremely feature rich product and may give users pause when getting started. This Day One book allows the user to get up and running quickly into the wonderful world of Junos."
Rob Cameron, Technical Marketing Engineering Manager, Data Center SRX, Juniper Networks

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

- Understand the different ways to manage the SRX Series Services Gateways.
- Console to a SRX device for the first time and start the initial boot process.
- Review operational and configuration modes.
- Review interfaces, security zones, and security policies.
- Configure basic IP connectivity and elements to enable remote access.
- Configure basic static routing.
- Upgrade the firmware of the SRX device.
- Configure different levels of local administration and external administrators.
- Configure additional network and system management resources.
- Write basic security policies.
- Configure different NAT source scenarios.
- Import the SRX devices into NSM.
- Enable different logging and troubleshooting tools.

Published by Juniper Networks Books

ISBN 978-1-936779-10-9

51800

9 781936 779109

7100 1265

JUNIPer NETWORKS

Junos® Dynamic Services Series

Day One: Deploying SRX Series Services Gateways

By Barny Sanchez

Published by Juniper Networks Books
Writers: Barny Sanchez
Editor in Chief: Patrick Ames
Copyediting and Proofing: Nancy Koerbel
Junos Program Manager: Cathy Gadecki

ISBN: 978-1-936779-10-9 (print)
Printed in the USA by Vervante Corporation

ISBN: 978-1-936779-11-6 (ebook)
Version History: v3 January 2011
3 4 5 6 7 8 9 10 #7100126

About the Author
Barny Sanchez (JNCIE FW/VPN #1, JNCIS-SSL, JNCIS-ER, JNCIS-M, JNCIS-SEC, JNCIA-IDP, JNCIA-AC, JNCIA-EX, JNCIA-WX, JNCIA-DX, JCNA, JNCI) holds a BS of Science in Information Systems Security from Westwood College, has completed advanced studies in Electronics Engineering from the Instituto Tecnologico de Costa Rica, and is currently undertaking a Master's degree in Information Assurance and Security at Capella University. He is a Consulting Engineer at Juniper Networks, specializing in Security Products and Solutions. Prior to this role, Barny worked as a Sr. Systems Engineer supporting Juniper Networks Strategic Partners, and before that, he spent over two years as a Senior Instructor, teaching most of Juniper's products. Before joining Juniper, Barny occupied management positions at different technical support organizations for Intel Corporation and Cisco Systems, in addition to spending several years designing and implementing multi-vendor networks for customers around the globe.

Author's Acknowledgments
The author would like to take this opportunity to thank Patrick Ames and Cathy Gadecki for helping to guide this book's progress, and to Senthil Nithiyananthan, Ryan Davis, and Brad Woodberg.

This book is available in a variety of formats at: www.juniper.net/dayone.
Send your suggestions, comments, and critiques by email to dayone@juniper.net.
Follow the Day One series on Twitter: @Day1Junos

What You Need to Know Before Reading this Book

Before reading this book you should have a basic understanding of the Junos operating system. Specifically, you should able to change configurations and to navigate through the command line hierarchy. You may reference other Day One books and free online training to help you acquire this background.

Other knowledge that will be helpful to you as you read through this book is:

- ✓ Understanding of TCP/IP.
- ✓ Knowing the difference between stateless and stateful firewall technologies.
- ✓ Familiarity with interface naming in devices running the Junos operating system.
- ✓ Although not mandatory to complete the reading of this book, access to SRX devices can help you practice configuring the various scenarios covered here, increasing the speed with which you implement SRX devices in your network.

The SRX

The SRX Series Services Gateway is a mouthful to pronounce. And the security device comes in several different platforms designed for a variety of networking uses. There are "small" SRX Series Services Gateways and there are "large" SRX Series Services Gateways.

This book simplifies the terminology by using the generic term *SRX*, or *the SRX*.

NOTE Some features of the SRX are configured differently on different platforms and this book attempts to point that out for you.

After Reading this Book, You'll Be Able to:

- ✓ Understand the different ways to manage SRX Series Services Gateways.
- ✓ Console to a SRX device for the first time and become familiar with the initial boot process.
- ✓ Review operational and configuration modes.
- ✓ Review interfaces, security zones, and security policies.
- ✓ Configure basic IP connectivity and elements to enable remote access via SSH, telnet, JWEB, NSM, etc.
- ✓ Configure basic static routing.
- ✓ Upgrade the firmware of the SRX device.
- ✓ Configure different levels of local administrators for full administration or read-only support.
- ✓ Configure external administrator authentication to better support a large number of administrators.
- ✓ Configure additional network and system management resources: NTP, SNMP, syslog logging.
- ✓ Write basic security policies to allow legitimate user data traffic and block unwanted traffic.
- ✓ Configure basic NAT source using the egress interface IP as the only source address.
- ✓ Configure basic NAT source using a pool of IP addresses.
- ✓ Import the SRX devices into NSM and perform basic routines.
- ✓ Avoid common mistakes when completing an initial configuration.
- ✓ Use commands and techniques to verify proper operation and to detect failures.

Want More SRX Info?

Chapter 1

Different Ways to Manage an SRX

The SRX is a very flexible security platform for performing configuration and management tasks because you can connect to it in multiple ways. Because it is a Junos® device, you have the option to manage a SRX via the console port, the command line interface (CLI), the web interface (J-Web), the Network and Security Manager (NSM), or through even more sophisticated mechanisms, like interfacing via scriptable tools. Some of these connection methods first require an initial configuration. Additionally, since every single configuration and management method is not available via all interfaces, it's important to understand what is possible with each method, then you can decide what is best for your environment.

This Day One book focuses on configuration and management via the Junos command line interface (CLI), but some tasks are shown via the J-Web interface and with the Network and Security Manager. This multi-dimensional approach should benefit your administration of the SRX.

Connecting Via the Console

Every SRX model, from the smallest to the largest, has a RJ45 console connection that is properly identified. So when connecting via the console all that is required is a rollover serial connection directly from your computer, or via a console server.

This connection method *does not* require any prior configuration on the SRX and the actual management interface is the CLI.

There are really no limitations on what you can do via this connection method, and for this reason it's considered perhaps one of the most powerful ways to manage the device.

To connect via the console you need: the supplied console cable, a computer with a serial port (or a USB to Serial adapter), and a terminal emulation application running on the connected computer.

Connecting Via the CLI

This method refers to a connection via a different port than the console, even though the actual interface is, again, the CLI, but it's different than the previous method because you use another port, and it requires some upfront configuration, the most significant being that an IP address and subnet mask are required.

What ports can you use? That's really up to your management requirements, but basically any port, or all ports, can be configured to accept management connections. Since you are now dealing with an IP connection, you can directly attach using a standard Ethernet patch cable, or you can also be thousands of miles away (provided, of course, that you can reach the configured IP address).

Typically, after doing an initial configuration via the console, most administrators configure the necessary elements to allow remote access to the device via protocols like SSH and telnet. There are several steps to enable remote access management via the CLI, and these are shown in subsequent chapters of this book.

Since the interface is the same as when you connect via the console, when managing the device you can enjoy the same unlimited power with some added benefits, the most important being the ability to accept multiple, concurrent administration connections, via the same ports.

NOTE The SRX3400/3600 and SRX5600/5800 also have a dedicated management port that is different than the console. This port is exclusive for management purposes, and if you have an out-of-band management network, then it is best to use these ports. Also known as *fxp0s*, these interfaces exist in the control plane of the SRX, and cannot be used for user data traffic (which helps to guarantee that the dedicated management channel remains open in the event that there is a disruption of the data plane).

Connecting Via the J-Web

J-Web is a powerful web-based interface that allows you to manage the SRX via a graphical interface on a web browser. This capability is available on all SRX platforms, big or small.

Connecting via the J-Web has similar requirements to connecting via the CLI. Some initial configuration needs to be done, such as setting an IP address and a subnet mask, as well as turning on this particular management service. As long as you can reach the IP address configured on any of the interfaces, and the corresponding services are turned on, then you can launch the graphical interface, even from miles away. Again, all the details are examined in a few more pages, so stay tuned.

While the J-Web has some limitations when it comes to enabling traceoptions (debugging), it features some pretty amazing graphics that can quickly help you determine the overall health of the system. As you become more versed with the CLI and its powerful syntax, you may switch back and forth between interfaces to explore how to configure different features of the SRX.

TIP Using the J-Web is sometimes the preferred connection method for administrators that are accustomed to managing other vendors' devices via graphical interfaces.

Connecting Via Network and Security Manager

Network and Security Manager (NSM) is Juniper Networks management platform for SRX firewalls. Generically referred as NSM, this management system is not limited to just SRX devices, and in fact it can potentially manage most of Juniper Networks portfolio, including routers, switches, secure access devices (SA), intrusion detection and prevention appliances (IDP), and more.

If you are planning to deploy dozens or even hundreds of SRX devices, then you must consider NSM. With it you can simplify provisioning tasks, and with just a few clicks you can push changes to the entire network, making what could be considered a network management nightmare, something easily completed.

Installation and operation of NSM warrants a whole Day One book to itself, or more likely, a whole series of Day One books, but there is specific Juniper Networks documentation and technical training to teach you what you need to learn about that subject. This book instead shows you how to import the SRX devices into NSM, if you already have NSM in your network, and then you can take it from there.

You'll need to first configure an IP address and management services into the SRX so that NSM can manage your firewall, and provided that this IP can be reached from the NSM server, you can then import and manage the firewalls exclusively through this tool.

In terms of limitations, the NSM is not suitable for tasks like debugging, but on the other hand, it is a good solution for logging events, and keeping your network in optimal condition. It is unquestionable that the power of NSM and all of its features far outweighs its few limitations.

MORE? There is so much more to NSM than what's discussed in these few paragraphs. If you want to learn more about it, start by reviewing the product specifics from the Juniper Networks website at www.juniper.net/us/en/products-services/network-management/. Training information is available at www.juniper.net/us/en/training/technical_education/.

Consoling to an SRX for the First Time

By now you may have realized that connecting via the console is unavoidable – that's because any other connection method that makes use of a port different than the console requires additional configuration to that chosen port in order to respond to management services.

This also holds true for the high-end SRX platforms (SRX3400/3600 and SRX5600/5800). These platforms do not have a standard number of interfaces right out-of-the-box, and given that they are modular devices, the administrator is required to connect one at least once via the console to commit an initial configuration enabling some form of remote access.

But the SRX branch portfolio devices (the SRX100, 210, 240 and 650) have a factory default configuration that enables an administrator to connect to them via some of the other methods right out-of-the-box.

NOTE Understanding the different aspects of the factory default configuration that enables this access requires you to study a few more missing pieces, but for now, let's focus on the console connection.

To connect via the console you need: the supplied console cable that came with the SRX device, a computer with a serial port (or USB to Serial adapter), and a terminal emulation application running on the computer.

To Console to an SRX for the First Time:

1. Connect the provided console cable to your computer, and at the other end, to the port of the SRX.

2. Open your terminal emulation program (such as Hyper Terminal in Windows).

3. In the application, set the port settings (COM port that identifies the serial connection) with the following information:

   - Bits per second: 9600
   - Data bits: 8
   - Parity: None
   - Stop bits: 1
   - Flow control: None

4. Click "Open" or "Connect" (the wording is application dependant).

TIP If you are an Apple user you're not out of luck. Connecting to the console of your device is just as easy. After connecting your USB-to-Serial adapter, find out what the name of the devices is ("$ ls /dev/"), and once you know this, open a terminal window and type "$ screen /dev/[device_name] 9600".

After connecting you might expect to see something on the screen. Depending on the stage of the boot process in which you connected, you may see a lot of information being displayed. The messages that appear are a normal part of the boot process, and these are really relevant when troubleshooting boot-related problems, or doing procedures like password recoveries.

Assuming that this is a brand new, or factory restored device, you should now have a login prompt like this (note the word Amnesiac, which is Juniper's way to indicate when a device has no defined configuration):

```
Amnesiac (ttyu0)

login:
```

In the event that the SRX has some configuration already committed (maybe this is not a *new* device), then the prompt is different. The prompt possibilities are many, depending whether the committed configuration has a banner or not, for example, but the key thing is that the word `Amnesiac` will not be there. An example:

```
srx210-1 (ttyu0)

login:
```

If you have made it this far and can't wait any longer, dive in! You may log in to the SRX by using the login name `root` and pressing [Enter] for the password. If you do you will see something similar to this:

```

--- JUNOS 10.1R1.8 built 2010-07-12 18:31:54 UTC

root@%
```

If you are unable to log in with the username `root` and no password, this means that the device has a different configuration than the factory settings. If you don't know the password of the root account, or of another account with super-user privileges, then a password reset is needed. The process to do a password recovery can be found here: http://kb.juniper.net/KB12167.

Very good! Let's get a cup of java or another favorite drink and get ready to delve a little deeper into the process. The objective of the next chapter is to review the command line interface, and the main components of the factory default configuration.

Chapter 2

Operational and Configuration Modes

Plenty of *Day One* books and additional online and printed resources cover general Junos operations, so you may be wondering why we do so again here.

Running Junos on a security device shifts the perspective from connectivity that *includes* security (unless strictly prohibited to) rather than *excludes* security (unless strictly permitted). This dramatic shift in posture means that Junos approaches connectivity in several different ways on the SRX Series. Junos makes any decision with security, not connectivity, as its primary objective.

For example, an EX Series or MX Series configured with an interface IP address and system services, immediately supports connectivity to other devices; while the SRX Series requires additional configuration steps.

So this chapter, covers not the details of operation and configuration modes that you are already know, but the aspects that set the SRX apart from M/MX/T routers and EX switches. (If you are wondering about the J-series routers, these started running the same Junos flavor available in SRXs since early 9.x versions.)

NOTE If you need to reinforce your knowledge of Junos operational and configuration modes, then see the *Junos Software Fundamentals Series* Day One books at www.juniper.net/dayone. Also seek out the *Juniper Networks Technical Library* at http://www.juniper.net/books.

Interfaces and Security Zones

The SRX services gateways (and J-series routers, for that matter) use a nested concept that consists of security zones. The implications of having zones are critical to Junos security architecture.

The first thing that administrators realize after configuring an IP address in an interface is that they cannot even ping it. This is because SRXs are locked-down devices, and no traffic is allowed in and out of the box without some prior configuration.

To permit traffic in and out of an interface, you need to configure more elements. Besides the IP address settings, an interface needs to be associated with a security zone, and the security zone in turn needs to be bound by association with a routing instance. This relationship is shown in Figure 2.1.

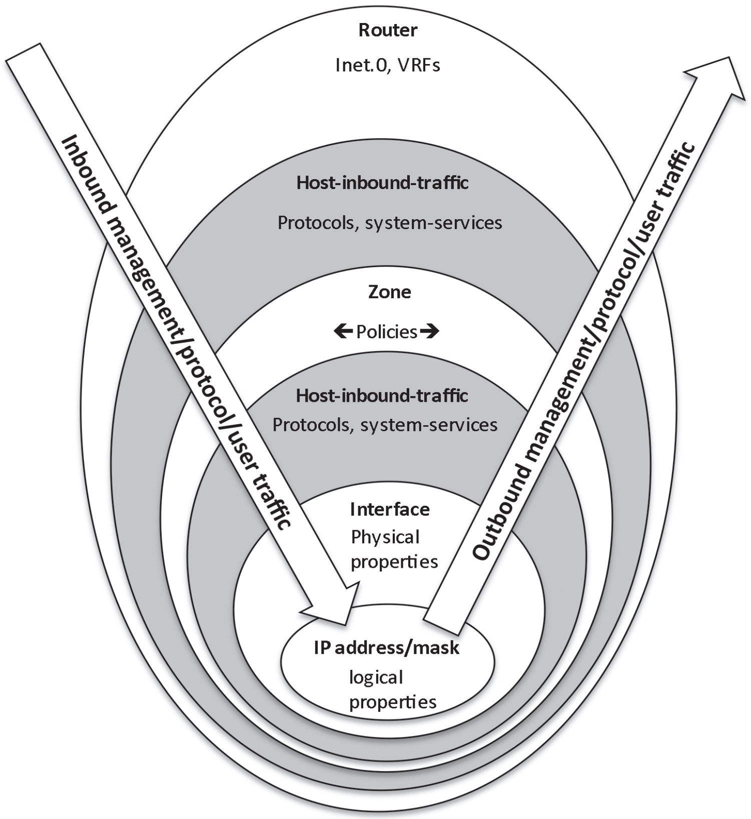


Figure 2.1 Interfaces, Zones, and Routing Instances

In Figure 2.1 you can see that traffic destined to an interface can be one of three different types:

- *management* (or system-services), for example ping and SSH.
- *protocol-related* such as OSPF and DHCP (or referred to simply as *protocol*).
- *user data traffic*, such as packets corresponding to the communication from a client to a server.

Notice that if traffic is destined to an interface for the purpose of managing the device, then this is where host-inbound-traffic comes into play. Host-inbound-traffic can be configured at two different levels. If configured at the zone level, then it will affect all the interfaces bound to that zone, but when configured at the interface level host-inbound-traffic will affect only that specific interface. In the event that host-inbound-traffic is configured at *both* the interface and zone levels, then the interface settings will take precedence. In other words, services are not added.

NOTE As you can see, this can be a little tricky. Let's try another example. If you configure ping system-services at the zone level, and try to ping an interface belonging to that zone, it will respond properly. If you later decide to enable telnet in the same interface by configuring system-services at the interface level, then ping will stop responding. This is because interface settings take precedence. To fix this and be able to get responses again, you need to enable ping system-services at the interface level.

Host-inbound-traffic settings have no effect at all over outbound traffic.

Before moving past the interface level, there are two important points you should note:

- First, configuring system-services host-inbound-traffic is not sufficient to manage a device via an interface. While the interface can accept those types of traffic, for some services like telnet, SSH, FTP, and J-Web access, you have to also enable corresponding services under `[edit system services]`.
- Second, host-inbound-traffic settings do not affect fxp0 interfaces, and they can not be configured for those interfaces. As discussed in Chapter 1, fxp0 interfaces are exclusively for management purposes, and as long as you configure an IP address, and turn on the services under `[edit system services]`, then you can connect remotely.

TIP If you want to apply policies to fxp0 interfaces and need to do things like permitting only certain subnets to connect, then you want to explore the SRX's use of filters. Keep this in mind, as the functionality is discussed soon.

A closer examination of Figure 2.1 also shows the presence of *policies*. The figure indicates that traffic from any zone, destined to any other zone (even if it's the same as the source), requires a security policy to be permitted to pass through. (Policies are not configured at the zone level, by the way.) The absence of security policies will simply black hole (drop) the traffic. This specific behavior is what it makes the SRX so secure by nature, but a little different than other Junos operating devices; you decide what specific traffic is permitted from one zone to another.

Be aware that traffic moving from any zone to any other zone, implies that this is user data or transiting traffic (referred to many times as *transient traffic*). Since management traffic is not traversing the firewall but is actually terminating at the firewall, then this does not require examination by any security policies.

Finally, the outer layer of Figure 2.1, the router, makes reference to the routing table instance. By default, all predefined zones or newly created zones are bound to the routing instance `inet.0` (IPv4 routing table), unless specified otherwise. Put differently, configuring an IP address in an interface, and binding it to a zone, creates a host routing entry table in `inet.0`.

So where do you configure all of these elements? Junos is fairly forthright here:

- Interfaces are configured under the `[edit interfaces]` hierarchy.
- Zones and host-inbound-traffic settings are configured under the`[edit security zones]` hierarchy.
- Policies are configured under the `[edit security policies]` hierarchy.
- And, system services are configured under the `[edit system services]` hierarchy.

The Factory Default Configuration

Now that you've digested a little background about zones and interfaces, you should be able to better understand the factory default configuration of the SRX.

First off, configuration is not the same for branch devices (SRX650 or lower) and the higher-end units (SRX 3400 or greater). This is because, typically, the product lines are positioned and configured for different scenarios, and, acknowledging this criterion, Juniper's engineers wanted to make the first-time installation as simple as possible.

By way of example, most SRX210 devices are found in branch offices belonging to a medium-to-large size organization, and the connection type is typically an Ethernet hand-off from the ISP that provisions a relatively small bandwidth, and DHCP for the CPE. On the other hand, SRX3400 clusters are typically found at a more central site, where the customer typically pays for more bandwidth and a dedicated pool of public IP addresses.

So, as a generalization, some of the things that the SRX factory default configuration features:

- Bootp services in the interface `ge-0/0/0`. Location: `[edit interfaces ge-0/0/0]/.`
- DHCP server services in the interfaces `ge-0/0/1 through ge-0/0/7`, with address allocation from the network 192.168.1.0/24. Location: `[edit system services dhcp].`
- Switched interfaces `ge-0/0/1` through `ge-0/0/7.` Location: `[edit interfaces interface-range interfaces-trust].`
- Security zones `trust` and `untrust`. Location: `[edit security zones].`
- Outbound Internet access using NAT with port address translation, permitting traffic from zones `trust` to `untrust`. Location: `[edit security nat].`
- General protection against any traffic sourced from the `untrust` zone. Location: `[edit security zones security-zone untrust].`
- System management services enabled. Location: `[edit system services].`
- Logging of critical events, as well as errors generated by commands typed by any administrator that attempts to change the configuration. Location: `[edit system syslog].`
- Basic configuration for managing Juniper's AX411 Wireless LAN Access Points. Location: `[edit wlan].`

The generalized environments and the more fixed physical nature of the smaller devices make the default configuration more predictable, and many times the factory default settings should be close to what you might need in terms of basic communication services. But read on, because the SRX goes way beyond "basic" in almost every capacity.

Conversely, the increased security requirements of the SRX3400s, along with their modularity, makes them more unpredictable to be able to suggest a highly practical factory default configuration, so the administrator here will find that the configuration for the high-end platforms is just a few lines long.

MORE? The Appendix contains samples of the entire factory configuration of an SRX210 and SRx3400 running Junos 10.1R1.8.

Try It Yourself: Examining the Factory Default Configuration

Display the different configuration blocks discussed in both this and the previous section (try alternating the commands below and making use of pipes | display set | no-more):

```
> show configuration
> show configuration security
> show configuration system services
> show configuration interfaces
```

Display the status of the interfaces and IP address assignment:

```
> show interface terse
```

Display the zones configured:

```
> show security zones details
```

Display a summary of nat source:

```
> show security nat source summary
```

Introducing a Work Topology

For the rest of this book, you should refer to the network topology shown in Figure 2.2 when you need a visual view of what is being demonstrated.

This book's testbed is that of an enterprise network, with a SRX210 acting as the firewall protecting resources in a branch office, and the SRX3400 as the head-end device, located at the corporate office. In a real world network, it is not uncommon to find as many as thousands of these branch devices connecting in the same way.

To keep things simple, let's assume there is a private and secure network between the devices (PRIVATE). This is not always the case as branch devices could have direct Internet access, and an IPSec tunnel or other mechanism can be configured to secure business communications throughout the entire network.

Also in Figure 2.2, note how the branch has Internet access via the head-end device.

Pay special attention to the delimitation of areas (trust, untrust, admins) as these highlight the concepts of zones presented earlier, and note in the case of the NSM and RADIUS servers, there is no zone allocation, since these services are connected via the fxp0 interface.

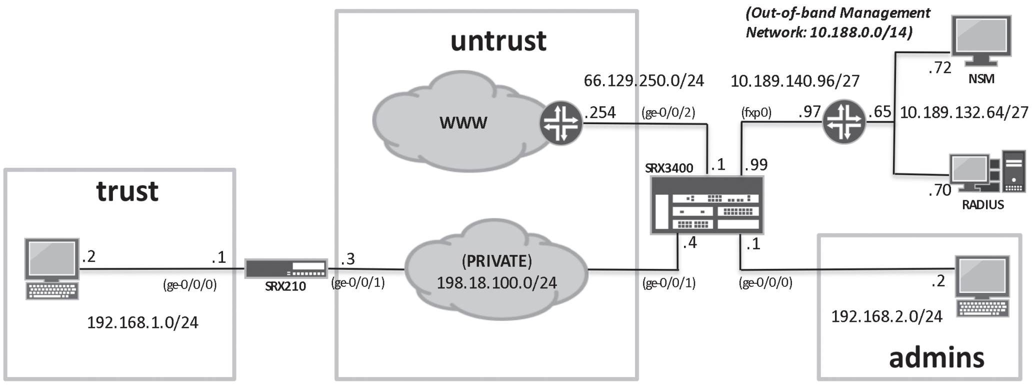


Figure 2.2 This Book's Network Topology

From a management perspective, it's not always the case that all services are directly attached to the fxp0 interface, and, like this sample network, an intermediate hop is used to reach a larger management network where many devices play an important role in the administration and monitoring tasks. Both 10.189.x.y/27 subnets belong to a larger network, 10.188.0.0/14. This is important to note since it can simplify the routing configuration.

Throughout this book we'll delve into more material, concrete scenarios and configuration steps that are based on this network topology and presented to you to try on your equipment. Let's get going, now, and establish remote access.

Chapter 3

Enabling Remote Access

Starting with enabling system services by configuring the interfaces and zones, this chapter puts the concepts covered so far into practice. Refer to the network topology shown in Figure 2.2 for a full understanding of the examples. Since the SRX3400 factory default is leaner, it gives us more to work on, and thus is the reason why most of the snippets in this section are from its platform.

Remember that in the case of SRX high-end platforms, the administrator has the additional option of using the fxp0 interfaces exclusively for management purposes. Because fxp0 interfaces exist in the control plane, and cannot be used for processing of user data traffic, there is no need to assign them to a zone security. However, if the administrator chooses to use a revenue port for management tasks, then all principles covered apply equally.

There isn't a specified order for configuring most aspects in Junos, however, in the forthcoming examples we'll use a best practice, as tested over the course of many, many devices, as the recommended approach.

Configuring System Services

The first best practice is to stop and think through what you want to accomplish before entering the device (this is true for any consoling method). So, to configure system services, we'll need to accomplish the following:

- Enabling SSH, telnet, FTP, and ping services.
- And, at the end of our configuration session, managing the box via any of the previously mentioned methods.

NOTE As stressed throughout this book, repetition is the mother of all learning: enabling system services is *not* enough to be able to manage a SRX using the configured services. Besides, with the exception of fxp0, as soon as an IP address is configured, you may manage the device via this interface.

To Configure System Services:

1. Connect via the console.

2. Login with root and enter configuration mode:

```
login: root
```

```
--- JUNOS 10.1R1.8 built 2010-02-12 17:24:20 UTC
root@% cli
root> edit
Entering configuration mode
[edit]
root#
```

3. Configure the system services:

```
[edit]
root# set system services ftp
[edit]
root# set system services ssh
[edit]
root# set system services telnet
[edit]
root#
```

4. Commit the changes:

```
root# commit
[edit]
  'system'
    Missing mandatory statement: 'root-authentication'
error: commit failed: (missing statements)
```

The error message you'll receive when trying to commit is expected for a new device that has no configuration, or one that has been factory defaulted, indicating that Junos requires the root user account to have a password or some sort of authentication. The easiest way to overcome this is by specifying a plain-text password.

5. Configure root authentication and commit:

```
root# set system root-authentication plain-text-password
New password:
Retype new password:
[edit]
root# commit
commit complete
```

No worries! Although the password you type is in clear text, Junos does a great job at encrypting it immediately, so that your configuration is not compromised if someone is looking over your shoulder.

NOTE By the way, the ping service was not forgotten. It is turned on by default for fxp0, but needs explicit configuration for all other interfaces, something that will is addressed in the next section. Also, ping cannot be activated under system services.

TIP If you are following along and trying these examples on your device, you may want to leave the console connection opened while reviewing the next section.

Configuring Interfaces and Zones

With system services now enabled, you may be tempted to try and connect remotely, but you are still unable to do so because you need to configure IP addresses on all the interfaces.

Let's review what you need to connect remotely:

- Configure the IP address and subnet mask in the SRX3400, per the diagram in Figure 2.2.
- Create the security zones admins, and untrust.
- Assign the interfaces to the corresponding zones.
- Turn on the telnet, FTP, SSH and ping on all interfaces of the SRX3400.

Before starting, take a moment to compare these configuration steps with the concepts presented in Chapter 2. Notice the alignment of these steps with the nested elements: an IP address goes into an interface, the interface into a zone, and a zone into a routing-instance. You do not configure custom routing-instances here, so the system assumes the default inet.0 (corresponding the IPv4 routing table). So the challenge is figuring out where the host-inbound-traffic takes its turn. (Response below.)

To Configure Interfaces:

1. While still connected to the console as root, enter configuration mode:

```
root> edit
Entering configuration mode
[edit]
root#
```

2. Start by configuring fxp0 since it is the simplest:

```
[edit]
root# set interfaces fxp0 unit 0 family inet address 10.189.140.99/27
[edit]
root#
```

3. Configure the interfaces ge-0/0/0, ge-0/0/1 and ge-0/0/2 using the same method:

```
[edit]
root# set interfaces ge-0/0/0 unit 0 family inet address 192.168.2.1/24
[edit]
root# set interfaces ge-0/0/1 unit 0 family inet address 198.18.100.4/24
[edit]
root# set interfaces ge-0/0/2 unit 0 family inet address 66.129.250.1/24
```

4. Commit the changes:

```
root# commit and-quit
commit complete
Exiting configuration mode
```

5. Verify that everything is working as expected:

```
root> show interfaces terse | match "fxp0.0|ge-0/0/0.0|ge-0/0/1.0|ge-0/0/2.0"
ge-0/0/0.0        up   up   inet    192.168.2.1/24
ge-0/0/1.0        up   up   inet    198.18.100.4/24
ge-0/0/2.0        up   up   inet    66.129.250.1/24
fxp0.0            up   up   inet    10.189.140.99/27
```

The output confirms that both the Admin and Link of the interfaces are operational.

To Configure Zones:

1. While still connected to the console as root, enter configuration mode:

```
root> edit
Entering configuration mode
[edit]
root#
```

2. Configure the untrust and admins security zones:

```
[edit]
root# set security zones security-zone untrust
[edit]
root# set security zones security-zone admins
```

To Bind the Interfaces to the Zones:

Reference Figure 2.1 and use the same set of commands to bind the interfaces to the corresponding zones:

```
[edit]
root# set security zones security-zone admins interfaces ge-0/0/0.0
[edit]
```

```
root# set security zones security-zone untrust interfaces ge-0/0/1.0
[edit]
root# set security zones security-zone untrust interfaces ge-0/0/2.0
```

BEST PRACTICES Always configure zones from the perspective of the firewall that you are setting, not from the perspective of the other devices in the network. For example, notice that in this example you do not need to configure the zone trust, as this is irrelevant and can even be unknown from the perspective of the administrator configuring the SRX3400. Also notice that both ge-0/0/1 and ge-0/0/2 belong to the same security zone. There are virtually no limits on what zones you may bind interfaces to, but an interface can only be bound to one zone at any given time.

To Enable SSH, telnet, FTP, and ping in the Interfaces:

SSH, telnet, FTP, and ping are all host-inbound-traffic system-services used for management purposes. Use the same set of commands to enable the desired services:

```
[edit]
root# set security zones security-zone untrust host-inbound-traffic ssh
[edit]
root# set security zones security-zone untrust host-inbound-traffic telnet
[edit]
root# set security zones security-zone untrust host-inbound-traffic ftp
[edit]
root# set security zones security-zone untrust host-inbound-traffic ping
[edit]
root# set security zones security-zone admins host-inbound-traffic ssh
[edit]
root# set security zones security-zone admins host-inbound-traffic telnet
[edit]
root# set security zones security-zone admins host-inbound-traffic ftp
[edit]
root# set security zones security-zone admins host-inbound-traffic ping
```

Do not forget to commit:

```
root# commit and-quit
[edit security]
  'idp'
    Failed to fetch the sec-db version
commit complete
Exiting configuration mode
```

The configuration committed with an error message – at this time routing has not been configured, and probably licensing has not been applied in your device either – as soon as these are configured, then this message will go away. (A description and sample of how to apply the licensing to an SRX is available in the Appendix.) If you do not have access to licensing at this time, or do not have a need to configure IDP services, then you can either ignore the message, or delete the hierarchy `[edit security idp]` and commit again.

BEST PRACTICES Acknowledging the fact that this is a document for security adventurists, telnet and FTP are inherently insecure protocols that offer no data protection. So, before turning these services on in your devices, analyze your need for them. If you absolutely have to configure them, try to do so on the internal side, never on interfaces facing public networks. In this instance we are setting aside best practices only for instructional purposes.

Configuring Basic Routing

From the perspective of the SRX3400 within our practice network (Figure 2.2) topology, the following routing needs are required:

Communication between the branch office over the private network can happen in one of many ways. For the purposes of this book, let's deploy NAT from and to both devices. Since the 198.18.100.0/24 network is directly connected, no additional routing configuration is required to reach any devices in this network.

The admins zone is a directly connected network segment. There are no real routing requirements to communicate to devices in that segment.

Communication towards the Internet, or untrust zone, occurs via the router with the IP address 66.129.250.254/24. So a default route is needed to reach all unknown networks via this device.

To communicate with devices in the out-of-band management network, routing is needed to send packets to the router with the IP address 10.189.140.97.

There are multiple ways to configure an environment like this, but let's keep things simple and configure access to the Internet and to the management network using static routing.

To Configure Access to the Internet:

1. While still connected to the console as root, enter configuration mode:

```
root> edit
Entering configuration mode
[edit]
root#
```

2. Configure a static default route pointing to the Internet router as the next hop:

```
[edit]
root# set routing-options static route 0/0 next-hop 66.129.250.254
[edit]
```

3. Configure a static default route to be able to reach the 10.188.0.0/14 management network:

```
[edit]
root# set routing-options static route 10.188.0.0/14 next-hop 10.189.140.97
[edit]
```

4. Commit and verify that the routes are active and everything appears to be in order:

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

root> show route

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

0.0.0.0/0          *[Static/5] 00:00:25
                   > to 66.129.250.254 via ge-0/0/2.0
10.188.0.0/14       *[Static/5] 2d 00:01:08
                   > to 10.189.140.97 via fxp0.0
10.189.140.96/27    *[Direct/0] 2d 00:01:08
                   > via fxp0.0
10.189.140.99/32    *[Local/0] 2d 00:01:08
                    Local via fxp0.0
66.129.250.0/24     *[Direct/0] 6d 15:32:16
                   > via ge-0/0/2.0
66.129.250.1/32     *[Local/0] 6d 15:32:16
                    Local via ge-0/0/2.0
192.168.2.0/24      *[Direct/0] 6d 15:32:16
                   > via ge-0/0/0.0
192.168.2.1/32      *[Local/0] 6d 15:32:16
```

```
                   Local via ge-0/0/0.0
198.18.100.0/24   *[Direct/0] 6d 15:32:16
                 > via ge-0/0/1.0
198.18.100.4/32   *[Local/0] 6d 15:32:16
                   Local via ge-0/0/1.0
```

This output generates a couple of observations of note. You can see that Junos automatically generates network and host entries to reach other hosts in the different segments that it is directly connected to. In addition, it inserts asterisks to indicate all the active routes – if by chance that does not happen on your device, double-check to ensure that your cables are connected properly.

At this point you should be able to ping the interfaces— as well as telnet, SSH, or FTP —only via directly connected devices. So a ping from the PC located in the zone admins to the IP address192.168.2.1 works, but it will not work from the same PC to the Internet-facing interface with the IP address 66.129.250.1. The reason? It's not routing but the lack of security policies that prevents you from crossing zones. This book covers security policies in Chapter 6, but don't skip ahead yet. Let's worry about some other fundamental management aspects, like setting up different administrators.

Now that you're connected, let's cover one last thing before leaving this chapter.

You'll want to transfer files in and out of the SRX, and FTP can facilitate this, but this protocol is inherently insecure: not only are the username/passwords sent via the wire in clear text but so is the data. Now that you have SSH enabled, you can use SCP (Secure Copy Protocol) for file transferring. Any Unix, Linux, or MacOS system has a native SCP application for this action. If you are a Windows user, you can download an open source application, such as WinSCP (http://winscp.net/eng/download.php).

For example, say you want to copy a new Junos install package from a Linux terminal to the SRX for the purpose of upgrading:

```
[barnys@server1 junos]$ scp junos-srx3000-10.1R1.8-domestic.tgz root@10.189 .140.99:~
The authenticity of host '10.189.140.99 (10.189.140.99)' can't be established.
RSA key fingerprint is 4c:21:ea:6a:fd:f5:b4:88:a4:61:ad:d5:fe:81:10:46.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.189.140.99' (RSA) to the list of known hosts.
root@10.189.140.99's password:
junos-srx3000-10.1R1.8-domestic.tgz                     100%  172MB   2.7MB/s   01:03
[barnys@server1 junos]$
```

You can confirm that the file was copied by using the Junos CLI:

```
root> file list

/cf/root/:
.cshrc
.history
.login
.profile
.ssh/
junos-srx3000-10.1R1.8-domestic.tgz*

root>
```

Chapter 4

Configuring Administrators

Only the root administrator can connect to the SRX via the configuration we've implemented. This can be problematic, especially if more than one person needs to connect to the SRX. Sharing the root account among multiple users creates several difficulties: there is no way to account for who really logged in and made changes (hopefully the changes won't disrupt network operations); this one account has unlimited powers, and there is virtually nothing the root account cannot do (any administrator can cause a lot of headaches if he or she inadvertently deletes or modifies the configuration); and troubleshooting and monitoring are an issue.

In this chapter let's configure local *and* remote administrator accounts with different privileges. Local accounts exist only in the device where they're configured, and this is ideal when a handful of individuals connect to the SRX. If dozens of administrators, or even more, connect to these devices, like in a NOC (Network Operations Center) or a large enterprise, then you are better off leveraging the centralized authentication mechanisms offered by RADIUS or TACACS+.

To demonstrate the flexibility of Junos, three local accounts will be configured with the following privileges:

- barnys (super-user)
- halle (read-only)
- and, max (operator)

A fourth account, carrie, has a more complex set of requirements, allowing her to change only interface settings.

The same requirements are configured using RADIUS. This book does not offer insight into the theory and operation of RADIUS, it simply explains how to configure the requirements presented. The Appendix, however, shows screenshots taken when the server used for the examples in this book was being configured, and is something that may be useful for the administrator that needs to integrate RADIUS with Active Directory. This is a trivial task using Juniper's Steel Belted Radius (SBR) product. For more details of this product refer to the following link: http://www.juniper.net/us/en/products-services/software/ipc/sbr-series/enterprise/.

NOTE Another option exists using TACACS+, but it is not discussed in this short book. If this is your only option, please refer to the SRX technical documentation at www.juniper.net/techpubs/.

Users and Classes

Junos has four predefined *user* accounts that cannot be renamed or modified: operator, read-only, super-user, and unauthorized. And a *class* is a container of permissions that defines allowed/denied commands and configuration options.

Perhaps the easiest user account to understand is super-user. By default, the user root belongs to this class, which is what grants this account unlimited reach and access.

The other classes define a more limited set of permissions. If the preset classes do not meet your requirements, then you can configure a custom class, and specify the mix of allowed commands and permissions for the users that belong to that class. This is precisely what you configure later for the user *carrie*.

A user can belong to only one class. If you think of this as a limitation, consider that Junos allows you to configure as many classes as you need. There is no set limit on the number of users and classes you can configure, so if you are relying on a centralized solution, then you can have even more flexibility.

Configuring Different Local Administrators

What steps are necessary to configure different local administrators?

- Configure the local accounts *max*, *halle*, *barnys*, and assign them to the local classes operator, read-only, and super-user, respectively.
- Create a local class consultant with restricted access to modify interfaces.
- Configure the local account *carrie* and assign this to the consultant class.
- Test to be sure that things are working as expected.

To Configure the Local Accounts:

1. Configure the users *max*, *halle* and *carrie* and assign them to their corresponding predefined classes:

```
[edit]
root# set system login user max class operator authentication plain-text-password
New password:
Retype new password:

[edit]
root# set system login user halle class read-only authentication plain-text-password
New password:
Retype new password:

[edit]
root# set system login user barnys class super-user authentication plain-text-password
New password:
Retype new password:
```

2. Create a local class *consultant* with restricted access to configure interfaces only:

```
[edit]
root# set system login class consultant allow-configuration interfaces
[edit]
root# set system login class consultant permissions configure
```

3. Configure the user *carrie* and assign this to the class *consultant*:

```
[edit]
barnys# set system login user carrie class consultant authentication plain-text-password
New password:
Retype new password:
```

Commit the configuration:

```
[edit]
barnys# commit
commit complete
```

VERIFY Take a moment to verify that your accounts are working as expected. Understanding what to expect from every class is critical to mitigating many management problems in your network. Here, only *max*, *halle* and *carrie* are verified as the account *barnys* is not any different than the root account used so far.

4. Test the user account *max*:

```
[barnys@server1 ~]$ ssh max@10.189.140.99
max@10.189.140.99's password:
--- JUNOS 10.1R1.8 built 2010-02-12 17:24:20 UTC
max> configure
     ^
unknown command.
```

```
max> show configuration
## Last commit: 2010-04-11 04:13:21 UTC by barnys
version /* ACCESS-DENIED */;
system { /* ACCESS-DENIED */ };
interfaces { /* ACCESS-DENIED */ };
routing-options { /* ACCESS-DENIED */ };
security { /* ACCESS-DENIED */ };

max> clear interfaces statistics all
max> traceroute 10.189.132.70
traceroute to 10.189.132.70 (10.189.132.70), 30 hops max, 40 byte packets
 1  10.189.140.97 (10.189.140.97)  1.011 ms  0.718 ms  0.654 ms
 2  10.189.132.97 (10.189.132.97)  0.684 ms  0.220 ms  0.207 ms
 3  10.189.132.70 (10.189.132.70)  1.650 ms  0.361 ms  0.355 ms

max> ping 10.189.140.97 count 2
PING 10.189.140.97 (10.189.140.97): 56 data bytes
64 bytes from 10.189.140.97: icmp_seq=0 ttl=64 time=0.873 ms
64 bytes from 10.189.140.97: icmp_seq=1 ttl=64 time=0.878 ms

--- 10.189.140.97 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.873/0.875/0.878/0.003 ms
```

Notice how the user *max* that was assigned to the class *operator* cannot make configuration changes (he simply cannot go into configuration mode), or read the configuration. He can, however, clear interface statistics, and run traceroutes, pings, or diagnostics commands. This is the behavior expected of this class.

5. Test the user account *halle*:

```
[barnys@server1 ~]$ ssh halle@10.189.140.99
halle@10.189.140.99's password:
--- JUNOS 10.1R1.8 built 2010-02-12 17:24:20 UTC
halle> configure
      ^
unknown command.

halle> clear
      ^
unknown command.

halle> ping
      ^
unknown command.

halle> show configuration
## Last commit: 2010-04-11 04:13:21 UTC by barnys
version /* ACCESS-DENIED */;
```

```
system { /* ACCESS-DENIED */ };
interfaces { /* ACCESS-DENIED */ };
routing-options { /* ACCESS-DENIED */ };
security { /* ACCESS-DENIED */ };

halle> show system uptime
Current time: 2010-04-11 04:34:02 UTC
System booted: 2010-03-29 14:30:13 UTC (1w5d 14:03 ago)
Protocols started: 2010-03-29 14:31:16 UTC (1w5d 14:02 ago)
Last configured: 2010-04-11 04:13:21 UTC (00:20:41 ago) by barnys
 4:34AM  up 12 days, 14:04, 2 users, load averages: 0.00, 0.00, 0.00

halle> show interfaces fxp0
Physical interface: fxp0, Enabled, Physical link is Up
  <snip>
```

The account *halle* is restricted to operational mode show commands. She cannot clear interfaces or run applications. The class *read-only* is good for administrators in charge of monitoring the device's operational status.

6. Test the user account *carrie*:

```
[barnys@server1 ~]$ ssh carrie@10.189.140.99
carrie@10.189.140.99's password:
--- JUNOS 10.1R1.8 built 2010-02-12 17:24:20 UTC
carrie> show
        ^
unknown command.
carrie> edit
Entering configuration mode
Users currently editing the configuration:
  barnys terminal p0 (pid 20718) on since 2010-04-11 04:10:24 UTC, idle 00:30:32
      [edit]

[edit]
carrie# show
## Last changed: 2010-04-11 05:02:53 UTC
interfaces {
    ge-0/0/0 {
        unit 0 {
            family inet {
                address 192.168.2.1/24;
            }
        }
    }
    ge-0/0/1 {
        unit 0 {
            family inet {
                address 198.18.100.4/24;
            }
        }
    }
    ge-0/0/2 {
        unit 0 {
            family inet {
                address 66.129.250.1/24;
            }
        }
    }
    fxp0 {
        unit 0 {
            family inet {
                address 10.189.140.99/27;
            }
        }
    }
}

[edit]
carrie# edit interfaces fxp0

[edit interfaces fxp0]
carrie# set description "Connects to 10.188.0.0/14 for management only"

[edit interfaces fxp0]
carrie# commit and-quit
commit complete
Exiting configuration mode
```

As expected, the user *carrie* is limited to viewing and modifying interface settings only.

TIP Now that every user has unique accounts, you can see exactly what the different administrators typed when they connected, something that was not possible if everyone was sharing the same root account. The factory default configuration has enabled the logging of interactive commands, and you can see the log with the `show log interactive-commands` command. This is a *very* powerful forensics tool.

Configuring RADIUS Support

To demonstrate the functionality of RADIUS and to avoid any confusion with the local accounts configured, you need to delete the users *barnys*, *max*, *halle*, and *carrie*:

```
[barnys@server1 ~]$ ssh root@10.189.140.99
```

```
root@10.189.140.99's password:

[barnys@server1 ~]$ ssh root@10.189.140.99
root@10.189.140.99's password:
--- JUNOS 10.1R1.8 built 2010-02-12 17:24:20 UTC
root@% cli
root> configure
Entering configuration mode

[edit]
root# delete system login user barnys

[edit]
root# delete system login user max

[edit]
root# delete system login user halle

[edit]
root# delete system login user carrie

[edit]
root# commit
commit complete
```

By the way, the user class *consultant* remains intact as you need it to for the radius authorization.

There are different approaches for the configuration of the RADIUS authentication and authorization components, and you should choose your approach based on your infrastructure and how versed you are in the RADIUS server side. Here are two approaches that will hopefully be of use.

The first approach consists of configuring the user classes locally in the SRX and using RADIUS exclusively for authentication purposes. This works well, but if there are hundreds or thousands of devices in your network, then you run into the risk of decentralizing the authorization process. So, for example, if you create the class *consultant* like in the previous example, and if by mistake you don't configure it exactly the same way in all of the devices, then you can cause a major headache. Make sure that you clearly understand that the authentication comes from the RADIUS server in this approach, whereas the authorization (what the user can do) comes from the locally defined classes.

The second approach consists of configuring and obtaining both authentication and authorization from the RADIUS server. While this seems to be the preferred and more powerful approach, the administrator has to be certain of how to configure the RADIUS server to grant authentication, and then pass the correct attributes back to SRX. The advantage of this is that with a few clicks in your RADIUS server and you can grant access to the desired classes to thousands of administrators.

Regardless of the approach you select, there are configuration elements that are the same, such as telling the SRX that RADIUS will be used for authentication, the IP address, the secret key, and any additional options. Let's try it.

To Configure Radius Support:

Configure the authentication order:

```
[edit]
root# set system authentication-order [radius password]
```

WARNING Don't lock yourself out! When you configure the authentication order *only* to RADIUS, this means that the local user database will only be checked if the SRX cannot establish a communication with the RADIUS server (i.e., server is down). When you configure the authentication order as in the example, this gives you the possibility of connecting with local user accounts (like root), in the event that the server can be reached, but your account is not properly configured in the RADIUS system. In other words, configure it as in the example, so you always have a back-door entry in case something goes wrong on the server side.

Configure the server. Remember that the secret password (in this case *secret123*) has to be shared by the SRX and the RADIUS server (refer to the Figure 2.2 for the location and information of the server):

```
[edit]
root# set system radius-server 10.189.132.70 secret secret123
```

Next you configure RADIUS authentication using the local user classes. This is in alignment with the first approach of configuring the user classes locally in the SRX and using RADIUS exclusively for authentication purposes discussed earlier.

To Configure Radius Authentication and Local Authorization:

1. Configure the user accounts, and assign them to the local classes:

```
[edit]
root# set system login user barnys full-name “Super-user rights” class super-user

[edit]
root# set system login user max full-name “Operator rights” class operator

[edit]
root# set system login user halle full-name “Read-only rights” class read-only

[edit]
root# set system login user carrie full-name “Consultant rights” class consultant
[edit]
root# commit
commit complete
```

From the perspective of the administrator, everything with this configuration should work as it did in the previous section. For example, if *halle* connects, her authentication will be granted from the radius (something that is totally transparent to her), and once at the command-line prompt, she will only be able to use only operational mode show commands (the authorization granted by the local class read-only).

Now let’s move on to the second approach. If you want to get both authentication and authorization from the RADIUS server, this simplifies the SRX configuration, but the server side gets a little more complicated, because you have to know the attributes that the server needs to pass back to the firewalls.

To Configure Radius Authentication and Authorization:

1. Delete all configured system login parameters:

```
root# delete system login

[edit]
```

Be clear about this last step: neither the previous user accounts nor the class consultant are needed any longer, since the RADIUS server provides both.

2. Create a user to be used as a connection template for all RADIUS users:

```
[edit]
root# set system login user remote full-name "Radius-user template" class unauthorized
```

3. Confirm the configuration and commit:

```
[edit]
root# show system login
user remote {
   full-name "Radius-user template";
   class unauthorized;
}

[edit]
root# commit
commit complete
```

The use of the class *unauthorized* here may seem strange. When you configure a new user you have to specify a class, otherwise Junos does not let you commit, and *unauthorized* is in essence a class with no permissions, since the actual permissions are being passed in the form of RADIUS attributes from the server. Notice what happens if you attempt to configure a user with no class definition:

```
[edit]
root# set system login user remote1 full-name "Radius-user
template2"

[edit]
root# show system login
user remote {
   full-name "Radius-user template";
   uid 2004;
   class unauthorized;
}
user remote1 {
   full-name "Radius-user template2";
   ## Warning: missing mandatory statement(s): 'class'
}

[edit]
root# commit
[edit system login]
  'user remote1'
   Missing mandatory statement: 'class'
error: commit failed: (missing statements)
[edit]
root# rollback 0
load complete
```

Believe it or not, this is all that's required for authorization of many, many devices – a single line of code can fulfill all of your requirements, but as anticipated, this increases the complexity of the server side.

Don't forget to refer to the Appendix if you are interested in the main configuration components of Active Directory and Steel-Belted Radius. Remember, there are many RADIUS server options available that may feel and look totally different than the procedures noted in the Appendix.

Chapter 5

Configuring Network and System Management

Now that the initial configuration of IP connectivity, including basic routing and setting different levels of administrators, is in place, you can configure additional network management options such DNS, NTP, and SNMP (Domain Name Server, Network Time Protocol, and Single Network Management Protocol).

Services like these can be configured at any time, but it only makes sense to configure them as soon as IP connectivity and routing are in place. Also, please note that in the case of the lower-end branch SRX devices, there are more configuration tasks required to enable revenue ports to handle these management services, something that at first seems counterintuitive until you realize the different configurations of the SRX platform.

MORE *Junos Security,* from O'Reilly Media, provides an excellent first chapter on the difference within the SRX Services Gateway platform. See www.juniper.net/books.

Configuring NTP

You are encouraged to configure NTP as soon as possible and for a variety of reasons. From the perspective of this book, it would be very beneficial to learn not only who the administrators that log in are, but also when they log in. NTP is necessary to stamp logs, which can help in troubleshooting tasks. Also, logging is often configured within various security policies, so that you can track the creation and tear down of sessions.

The process of configuring NTP is straightforward, and consists basically of telling the SRX where the NTP server is, adjusting the connection parameters (optional), changing the clock, and testing that the time is being updated properly.

To Configure NTP Support:

1. Configure the NTP server and time zone:

```
[edit]
root# set system ntp server 64.90.182.55

[edit]
root# set system ntp boot-server 64.90.182.55

[edit]
root# set system time-zone America/New_York
```

NOTE If you don't have an NTP server already in place in your local network, you can use a publicly available one. For a reference to public NTP servers visit the following NIST website: http://tf.nist.gov/tf-cgi/servers.cgi.

Note that the difference in our NTP configuration is that the boot-server option is only referenced by Junos upon boot-time. Once the system has fully restored, then it uses the other server specified in the first entry above. So these servers *can* be different, although they are not in this example.

2. Adjust the connection parameters (optional). Let's first see what they are:

```
root# set system ntp ?
Possible completions:
  <[Enter]> Execute this command
+ apply-groups         Groups from which to inherit
configuration data
+ apply-groups-except  Don't inherit configuration data from
these groups
> authentication-key   Authentication key information
  boot-server          Server to query during boot sequence
> broadcast            Broadcast parameters
  broadcast-client     Listen to broadcast NTP
> multicast-client    Listen to multicast NTP
> peer                 Peer parameters
> server               Server parameters
  source-address       Use specified address as source address
+ trusted-key          List of trusted authentication keys
  |                    Pipe through a command
```

As you can see, Junos provides options for NTP authentication, source-addresses, and more. In most scenarios this is something that you do not have to worry about, but the options are available later on if you need them. We'll leave it be for now.

3. Update the system clock to make use of the new NTP server settings:

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

root> set date ntp
10 Jun 01:57:36 ntpdate[2730]: step time server 64.90.182.55 offset -0.000381 sec
```

4. Verify that the date and time were updated properly:

```
root> show system uptime
Current time: 2010-06-10 01:58:20 EDT
System booted: 2010-06-09 14:31:27 EDT (11:26:53 ago)
Protocols started: 2010-06-09 14:32:38 EDT (11:25:42 ago)
Last configured: 2010-06-10 01:57:33 EDT (00:00:47 ago) by root
 1:58AM  up 11:27, 1 user, load averages: 0.20, 0.08, 0.02
```

5. Verify the status and associations of the NTP:

```
root> show ntp status
status=c035 sync_alarm, sync_unspec, 3 events, event_clock_reset,
version=”ntpd 4.2.0-a Fri Feb 12 17:04:57 UTC 2010 (1)”,
processor=”powerpc”, system=”JUNOS10.1R1.8”, leap=11, stratum=16,
precision=-18, rootdelay=0.000, rootdispersion=1.215, peer=0,
refid=STEP, reftime=00000000.00000000  Thu, Feb  7 2036  1:28:16.000,
poll=4, clock=cfbafda2.3f93aff1  Thu, Jun 10 2010  1:58:58.248, state=3,
offset=0.000, frequency=0.000, jitter=0.004, stability=0.000

root> show ntp associations
    remote           refid  st t when poll reach   delay   offset  jitter
==============================================================================
*64.90.182.55    .ACTS.      1 -    1   64    1   14.081    0.921   0.293
```

Configuring DNS

Having DNS configured is essential for creating rules, when trouble-shooting, and also so that automatic updates and downloads of different SRX services, like IPS, work properly. Now is a good time to get it done.

You can enable the SRX to perform DNS lookups during the configuration process, but it’s also possible to specify the domain name and an appendix, so that you can resolve hosts in your network without fully qualifying them.

To Configure DNS Services:

1. Configure one or more DNS servers. They can be internal or external to the network:

```
[edit]
root# set system name-server 10.189.132.70
[edit]
root# set system name-server 10.189.132.68
```

2. Configure a domain name for the SRX:

```
[edit]
root# set system domain-name camlab.juniper.net
```

3. Configure a domain prefix to search in order to resolve local hosts without having to fully qualify them:

```
[edit]
root# set system domain-search camlab.juniper.net
```

4. Commit and test (refer to Figure 2.2 for internal device names):

```
[edit]
root# commit
commit complete

[edit]
root# run ping count 3 juniper.net
PING juniper.net (207.17.137.239): 56 data bytes
64 bytes from 207.17.137.239: icmp_seq=0 ttl=52 time=100.416 ms
64 bytes from 207.17.137.239: icmp_seq=1 ttl=52 time=100.566 ms
64 bytes from 207.17.137.239: icmp_seq=2 ttl=52 time=100.386 ms

--- juniper.net ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 100.386/100.456/100.566/0.079 ms

 [edit]
root# run ping count 3 radius
PING radius.camlab.juniper.net (10.189.132.70): 56 data bytes
64 bytes from 10.189.132.70: icmp_seq=0 ttl=126 time=0.441 ms
64 bytes from 10.189.132.70: icmp_seq=1 ttl=126 time=0.538 ms
64 bytes from 10.189.132.70: icmp_seq=2 ttl=126 time=0.828 ms

--- radius.camlab.juniper.net ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.441/0.602/0.828/0.164 ms

[edit]
```

NOTE Notice that in this case, DNS resolution is tested for both an external host (juniper.net) and an internal one (radius). In order for the internal resolution to work, then, there should be a DNS server (that is not shown in Figure 2.2).

NOTE In addition, if you omitted the previous Step 3 in the configuration, and wanted to ping internal hosts, you would have to fully qualify these hosts – for example, “ping radius.camlab.juniper.net”. This extra typing is totally unnecessary work if you are in the firewall constantly doing troubleshooting procedures.

Configuring SNMP

SNMP configuration and operation can be complex, but here we'll configure the following protocol because you can instruct the SRX to send notifications (traps) when different events occur, and client systems can be configured to connect to the SRX to poll for specific information at any time.

SNMP configuration in the SRX is straightforward, with a general section specifying device information and community, followed by more granular sections that specify trap options (which events trigger notifications), finishing with the client devices that are allowed to poll the SRX and SNMP servers.

To Configure SNMP Support:

1. First configure the device information and a community with read-only capabilities:

```
[edit snmp]
root#
[edit snmp]
root# set name SRX1
[edit snmp]
root# set location Cambridge
[edit snmp]
root# set contact “Barny Sanchez”
[edit snmp]
root# set community management authorization read-only
[edit snmp]
root# set community management clients 10.189.132.64/27
```

NOTE The “clients” are the management stations in the network that are allowed to poll the SRX. If you have a dedicated out-of-band network for management purposes, using a network subnet is very convenient. For extra security you can also specify individual IP addresses, and Junos will, in turn, interpret these as /32 or host devices.

2. Set trap options, or the IP that will be the source of the SNMP updates:

```
[edit snmp]
root# set trap-options source-address lo0
```

NOTE Using a loopback interface is a best practice. If you make this a habit across all devices, then you will have a consistent view of what devices generated the traps. This makes parsing tasks easier and can simplify the reporting generated from the network.

3. Configure the SNMP version, port, and which events will generate updates (use ? to view the different categories available):

```
[edit snmp]
root# set trap-group management version v2
[edit snmp]
root# set trap-group management destination-port 162
[edit snmp]
root# set trap-group management categories startup
[edit snmp]
root# set trap-group management categories authentication
[edit snmp]
root# set trap-group management categories services
[edit snmp]
root# set trap-group management categories link
```

4. Configure the target server, or the management station that will receive the generated traps:

```
[edit snmp]
root# set trap-group management targets 10.189.132.80
```

5. Confirm the configuration:

```
[edit snmp]
root# show
name SRX1;
location Cambridge;
contact “Barny Sanchez”;
community management {
   authorization read-only;
   clients {
      10.189.132.64/27;
   }
}
trap-options {
   source-address lo0;
}
trap-group management {
   version v2;
   destination-port 162;
   categories {
      authentication;
      link;
      startup;
```

```
         services;
      }
      targets {
         10.189.132.80;
      }
   }
```

Configuring Syslog

System logging has to do with internally generated events, bound to the control plane of the SRX. NTP, authentication services, chassis events and many more generate these kinds of events.

NOTE Our focus in this section will be on system logging. Security logging, when you configure security policies, will be covered in the next chapter. Security logging refers to the messages generated from matching a security policy, and whether the policy has logging enabled. These logs refer to events generated at the data plane after processing user data traffic.

To Configure Syslog Logging:

1. Configure the destination server, or event collector, along with any of the facility and severities desired:

```
[edit]
root# set system syslog host 10.189.132.70 source-address 10.189.140.99
[edit]
root# set system syslog host 10.189.132.70 any any
```

NOTE The source-address can be anything, but it is a good practice to specify either a loopback interface IP address, or the address of the egress interface for the events. This gives you consistency when reading and parsing through your log files.

Also, the value `any` was used to specify "any facility and severity value." For details on specific values, and what they mean, please refer to the RFC5424: http://tools.ietf.org/search/rfc5424#section-6.2.1. In the case of Junos, you can simply press [TAB] after keying in the host IP to see the list of facilities and severities available.

If you are following these configuration examples on a branch SRX device, you are probably wondering about the system syslog settings within the factory configuration. If so, see the section *SRX Default Factory Configurations* in the Appendix.

Junos enables logging to files stored in flash memory, but you can mix this with external logging of any events that you choose, or send all logging to an external device. This flexibility is great if you need to send certain events to specific servers.

Even in the factory configuration, Junos provides knobs to reduce the number and size of the files used for local logging, so you don't have to worry about filling up the storage memory. The following example shows a mix of logging to local files and FTP and authorization events being logged to different syslog servers:

```
[edit system syslog]
root# show
archive size 100k files 3;
host 10.189.132.70 {
   authorization warning;
}
host 10.189.132.72 {
   ftp info;
}
file messages {
   any critical;
   authorization info;
}
file interactive-commands {
   interactive-commands error;
}
```

Internal logs can be seen at any time with the command:

```
root> show log [filename]
```

The logs are simple text files, so it is possible to use pattern matching or other Junos search options to narrow your searches when looking for something.

Finally, external logs need to be seen at the target server, or via graphical console, if one is available.

Chapter 6

Writing Basic Security Policies

Security policies are at the heart of any of the firewall functions of the SRX Services Gateway platform. By default, traffic entering an interface destined to any address is going to be blocked. This is the expected default behavior, and no traffic is allowed through the SRX until you permit it to enter by using security policies.

NOTE An exception to this blocked traffic rule is the traffic in and out of the fxp0 (management) interface. This interface is an exception because it resides in the control plane of the device, and it cannot be used for user data traffic.

Policy configuration entitles an IF-THEN-ELSE algorithm: IF traffic X is matched, THEN action Y is performed, ELSE drop packet (default behavior).

Matching traffic (IF statement) consists of looking at packets for the five following elements:

- **Source zone:** the predefined or custom zone created from the perspective of the SRX that you are configuring.
- **Source IP:** any IP address, or an address book, that specifies a host IP, or a subnet. The source selected has to match the source zone.
- **Destination zone:** predefined or custom zone created from the perspective of the SRX that you are configuring.
- **Destination IP:** any IP address, or an address book that specifies a host IP, or a subnet. The destination selected has to match the destination zone.
- **Application:** predefined or custom service that defines the source/destination ports, protocol involved, and timeout value.

If an incoming packet matches all the previous five elements, the action (THEN statement) defines what to do with this or any other packets matching the same combination:

- deny: drops the packet (silently).
- reject: drops the packet and sends a TCP-Reset to the originator of the traffic.
- permit: permits the packet.
- log: instructs the SRX to create a log entry for matching packets.
- count: provides accounting information per session.

There can be many dozens —or even thousands— of policies configured in various SRX devices (this number varies by platform). When packets ingress any of the devices, they are evaluated against security policies in a top-down fashion until all five elements presented before are matched. If a match is found then the SRX does what it was instructed to do with those packets and stops evaluating through the rest of the policies. If the evaluation process reaches the last policy and no match was made (ELSE statement), then the default firewall action *deny-all* is applied.

Since the evaluation of firewall policies happens sequentially in a top-down manner, it is a rule of thumb to place the most specific rules at the top of the list and the most generic policies at the bottom. If you fail to do so, then you may shadow a more specific match criteria with a more common one.

Let's put all of this into practice with an example (again, refer to Figure 2.2's illustration of our book topology to fully understand the following requirements in the SRX3400). As always, let's first stop and think about what it is we're going to do before we start our configuration:

- Permit any traffic from any hosts in the zone "admins" to any destination in the "untrust" zone.
- Permit custom traffic from any hosts in the zone "admins" to any other host in the same zone.
- Deny any traffic from the zone "untrust" to "admins."

About Zones

To review, a zone is a logical container used to group interfaces with similar security requirements (see Figure 2.1). For example, assume that in your organization there is a Human Resources department, so all firewall ports assigned to HR can be bound to the zone "HR." All firewall interfaces used by Finance can be bound to a zone "Finance," and so on. Zone names are locally significant, and you can name them anything that makes the most sense to you.

TIP If you are working in a large deployment involving managed services or multiple groups, it is best that you use a structured naming convention, as this reflects in the logging that the firewall generates, making troubleshooting and accounting tasks simpler and cleaner.

There is a predefined security zone in the SRX called *junos-globlal* that can't be modified. In addition, the factory default configuration of branch SRX appliances has two additional zones: *trust* and *untrust*. These two other zones can be modified or even deleted.

Since the zones "admins" and "untrust" don't exist, these need to be created, and the corresponding interfaces then need to be assigned to them.

If you have been following this book sequentially from the beginning, you configured zones and interfaces in Chapter 3. The output should look like this:

```
[edit security]
root# show
zones {
   security-zone untrust {
      host-inbound-traffic {
         system-services {
            ssh;
            ftp;
            telnet;
            ping;
         }
      }
      interfaces {
         ge-0/0/1.0;
         ge-0/0/2.0;
      }
   }
   security-zone admins {
      interfaces {
         ge-0/0/0.0;
      }
   }
}
```

In addition, other commands can provide additional output of the zones available. Try the `show security zones ?` command and see for yourself.

Configuring Address Books

An *address book* is a name given to an IP address or prefix. The name you use is locally significant, and just like zone names, it pays to be organized with a naming convention when you are working with hundreds of zones in large deployments.

Unlike some network equipment from other vendors, when configuring Junos security policies in a SRX device, you reference the address books as the source *and* destination addresses, not as the actual subnets or IPs. This makes for a refreshing change, allowing you to set your configuration in a more human, readable format.

If you are in a hurry to move on to the next steps, you can always use the predefined address book *any* as the source and destination. This references *any and all* IP addresses.

Following our network topology in Figure 2.2, you are going to configure a specific address book for the computer in the zone "admins."

To Configure Address Books:

Stop and think for a moment, please! From the perspective of the firewall, where does the host reside? In this case it would be the "admins" zone, so make sure the address book is created in that zone.

Create the address book entry "PC1":

```
[edit security]
root# edit zones security-zone admins

[edit security zones security-zone admins]
root# set address-book address PC1 192.168.2.2
```

Verify the entry and notice how Junos automatically assumes this is a /32 entry since you didn't specify a subnet mask:

```
[edit security zones security-zone admins]
root# show
address-book {
   address PC1 192.168.2.2/32;
}
interfaces {
   ge-0/0/0.0;
}
```

Since the destination address can be any host in the "untrust" zone (the public web), it actually makes sense in this case to have used *any* as the address book for the destination address.

TIP Creating multiple address books for hosts in a zone is not a problem. However, if you want to gather those individual entries in a group to simplify your policy creation, then you can create *address-sets* for this purpose. The concept is very similar to the creation of an address book, but instead of specifying IP addresses or prefixes, you specify the individual address books that you want to belong to that address-set group. For example:

```
[edit security zones security-zone admins]
root# set address-book address-set PCs address PC1

[edit security zones security-zone admins]
root# set address-book address-set PCs address PC2
```

Configuring Services

Services specify the applications that we are matching, a combination of source/destination ports, protocol, and timeout. The ports and protocol are part of the TCP/IP packet header, and the timeout refers to the time that a particular packet will be held in memory before it is purged, if no subsequent packets match the same security policy.

The SRX Services Gateway devices are stateful firewalls. When an incoming packet is matched and an action is taken, then an entry identifying this packet and the corresponding action is held in memory (session table) so that subsequent packets are processed faster. If, after a while (the timeout value), no subsequent packets match the same criteria, the entry is purged from memory. A finite amount of entries can be held in memory, and that is why the firewall has to be judicious about what is held there.

You don't always have to configure services. In fact, there is a list of pre-defined services, and you can see their details from configuration mode with the command:

```
[edit]
root# show groups junos-defaults applications
```

If for some reason you need to create a service to accommodate a particular application in your network, then the process is to gather up the ports and protocol involved, consider the service timeout and make a decision about what it is, and configure a new service. For example, let's assume you want to configure a custom service "SERVICE1" to accommodate traffic that matches the following criteria:

- Source ports: range from 2000 to 4000
- Destination port: 1111
- Protocol: TCP
- Timeout: 1800 seconds

To Configure a Custom Service:

1. One single line of code allows you to create a new service. If you're following on your device, try keying in the following. Note that there are many options available to you as you key each command segment. Use the ? prompt as you are typing the following command:

```
[edit]
root# set applications application SERVICE1 source-port 2000-4000 destination-port
1111 protocol tcp inactivity-timeout 1800
```

2. Verify the application configuration:

```
root# show applications
application SERVICE1 {
   protocol tcp;
   source-port 2000-4000;
   destination-port 1111;
   inactivity-timeout 1800;
}
```

You can configure many custom applications; there is no hard-set limit in the number of applications that can be created.

Just like in the address-set option, it is possible to group multiple services together by configuring service-sets, making the configuration of security policies easier on you. The use of predefined and custom services is permitted, for example:

```
[edit]
root# set applications application-set MYSERVICES application SERVICE1
[edit]
root# set applications application-set MYSERVICES application junos-http
[edit]
root# set applications application-set MYSERVICES application junos-ping
[edit]
root# set applications application-set MYSERVICES application junos-dns-udp
[edit]
root# set applications application-set MYSERVICES application junos-dns-tcp
```

Configuring Security Policies

Configuring a security policy glues all the previous components together. Just remember that from the perspective of the policy, traffic is entering a zone and leaving a zone. Traffic can be entering and exiting interfaces bound to the same zone (intra-zone traffic), or distinct zones (inter-zone traffic) – regardless of the source and destination zones, the traffic will be dropped, by default mind you, if you don't configure a policy to tell it otherwise.

The combination of the from-zone and destination-zone is a context. When traffic arrives at an interface the SRX immediately knows the incoming zone as this is determined by the interface-zone binding. The destination zone, is determined by performing a route lookup of the destination IP address. Once the context is determined, the SRX then performs an ordered lookup in a top-down fashion of the policies trying to find a match for the packet. If a match is found, the corresponding configured action is performed, if not, the *deny-all* default is applied. Figure 6.1 can help you better understand this flow.

TIP For a detailed explanation of processing packets through the SRX, please refer to the *Junos Security Configuration Guide*. Version 10.1 can be located at: https://www.juniper.net/techpubs/software/junos-srx/junos-srx10.1/index.html. The book *Junos Security*, published by O'Reilly Media, also contains excellent descriptions and examples of the processing of packets through the SRX.

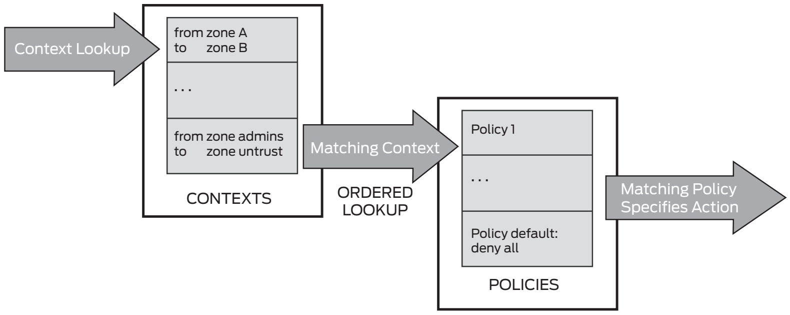


Figure 6.1 Context and Policy Lookup

Let's go back to the requirements presented at the beginning of this chapter, and configure a security policy:

Permit any traffic from any hosts in the zone "admins" to any destination in the "untrust" zone.

To Configure the Security Policy:

1. Identify the context:

```
[edit]
root# edit security policies from-zone admins to-zone untrust
```

2. Create a policy, giving it a name that makes sense. In this case, you are asked to create a policy to match any source/destination and

application. Notice that this only corresponds to the IF clause:

```
[edit security policies from-zone admins to-zone untrust]
root# set policy admins_to_untrust match source-address any destination-address any
application any
```

3. Now configure what to do (THEN clause) if the previous conditions are matched in an incoming packet. The requirement is to permit the traffic:

```
[edit security policies from-zone admins to-zone untrust]
root# set policy admins_to_untrust then permit
```

At this time you have already completed the first requirement. This was a very simplistic example, but you can take additional *then* actions, such as enabling counting for accounting purposes, or logging of events when a session is created, or closed, after matching this policy.

After enabling some of these other options the policy could end up looking like this:

```
[edit security policies from-zone admins to-zone untrust]
root# show
policy admins_to_untrust {
   match {
      source-address any;
      destination-address any;
      application any;
   }
   then {
      permit;
      log {
         session-init;
         session-close;
      }
      count;
   }
}
```

Let's now configure the second requirement from the list at the beginning of this chapter:

Permit custom traffic from any hosts in the zone "admins" to any other host in the same zone.

To Configure the Second Security Policy:

1. Identify the context:

```
[edit security policies from-zone admins to-zone untrust]
root# up
```

```
[edit security policies]
root# edit from-zone admins to-zone admins

[edit security policies from-zone admins to-zone admins]
```

2. Create a policy, giving it a name that makes sense. In this case, you are asked to create a policy to match custom traffic from any hosts in the same zone. For the custom traffic you can use the custom MYSERVICES services-set, configured previously:

```
[edit security policies from-zone admins to-zone admins]
root# set policy intra_zone_traffic match source-address any destination-address any
application MYSERVICES
```

3. Now configure what to do (THEN clause) if the previous conditions are matched in an incoming packet. The requirement is to permit the traffic, but you can also enable logging and counting:

```
[edit security policies from-zone admins to-zone admins]
root# set policy intra_zone_traffic then permit

[edit security policies from-zone admins to-zone admins]
root# set policy intra_zone_traffic then log session-init

[edit security policies from-zone admins to-zone admins]
root# set policy intra_zone_traffic then log session-close

[edit security policies from-zone admins to-zone admins]
root# set policy intra_zone_traffic then count
```

The second policy requirement has been fulfilled. As per the third requirement, there is no configuration needed, since the default policy *deny-all* will activate when the firewall determines the context of the traffic (from-zone untrust to-zone trust).

Verifying Security Policies

There are multiple ways to verify security policies. Visual inspection is one way, and is a good double-check that you don't have any typos, or unintended elements, in your configuration.

Try It Yourself: Examining Security Policies

Use `show` commands to verify the policies configured. Pay close attention to the output. The `detail` option is a favorite since it displays policy statistics.

```
> show configuration security policies
> show security policies ?
> show security policies to-zone untrust
> show security policies detail
```

Another way to verify that the security policies are working as expected is to test data traffic. In a production network, you can also inspect the session table that the SRX creates and match for specific information.

After logging into the PC at the “admins” zone and initiating multiple traffic sessions, inspect what the SRX is keeping in the session table with `show` commands:

```
[edit security policies]
root# run show security flow session
Session ID: 100001782, Policy name: admins_to_untrust/4, Timeout: 1796
  In: 192.168.2.2/4777 --> 216.52.233.201/443;tcp, If: ge-0/0/0.0
  Out: 216.52.233.201/443 --> 192.168.2.2/4777;tcp, If: ge-0/0/2.0

Session ID: 100001790, Policy name: admins_to_untrust/4, Timeout: 1800
  In: 192.168.2.2/4781 --> 209.239.112.126/80;tcp, If: ge-0/0/0.0
  Out: 209.239.112.126/80 --> 192.168.2.2/4781;tcp, If: ge-0/0/2.0

Session ID: 100001846, Policy name: admins_to_untrust/4, Timeout: 1404
  In: 192.168.2.2/4788 --> 66.235.120.98/80;tcp, If: ge-0/0/0.0
  Out: 66.235.120.98/80 --> 192.168.2.2/4788;tcp, If: ge-0/0/2.0

Session ID: 100002381, Policy name: admins_to_untrust/4, Timeout: 6
  In: 192.168.2.2/47621 --> 24.47.122.98/43519;udp, If: ge-0/0/0.0
  Out: 24.47.122.98/43519 --> 192.168.2.2/47621;udp, If: ge-0/0/2.0
<snip>
```

Every new packet that matches the criteria of a policy with an action of *permit* will generate an entry in this table. You can confirm the source/destination IPs, ports, protocol, time-out values, interfaces involved, the policy that matched the traffic, and more. Notice that entries are grouped in three lines at the time, representing the bidirectional information of a session (a flow). As traffic traverses, the SRX creates bidirectional entries in memory to allow the return traffic back through (this is what is meant by *stateful behavior*).

In a production network using high-end SRXs, it is necessary to keep track of millions of sessions, so issuing this command could render too much information. To narrow your search, you can use different parameters to search for traffic destined to a particular IP, or port, or coming from a given interface. The Junos version 10.1 offers the following options:

```
root# run show security flow session ?
Possible completions:
  <[Enter]>   Execute this command
  application   Show session for specified application or application set
  destination-port Show each session that uses specified destination port
  destination-prefix Show each session that matches destination prefix
  idp       Show IDP sessions
  interface   Show each session that uses specified interface
  protocol    Show each session that uses specified IP protocol
  resource-manager Show resource-manager sessions
  session-identifier Show session with specified session identifier
  source-port Show each session that uses specified source port
  source-prefix Show each session that matches source prefix
  summary   Show summary of sessions
  tunnel  Show tunnel sessions
  |       Pipe through a command
```

Logging of Security Events

Chapter 5 mentioned that the SRX could generate syslog messages for system events, and also for security events. Security events are generated when traffic matches a policy that has logging enabled.

The policies configured earlier in this chapter have logging enabled for `session-init` and `session-close`. This generates events when sessions are created and closed. In most cases, it is just enough to log on `session-close`, which will greatly reduce the amount of syslog traffic. But these extra entries in the security policy configuration are not enough to do logging.

Logging behaves differently in the branch SRX platform and the high-end data center SRX devices due to their hardware architecture. Although both device platforms have data and control planes, the high-end security devices make this division in hardware: given the limited resources in the control plane and the high number of entries that these devices can potentially generate, it's an important consideration when configuring security logging in the high-end platforms. The high-end

SRXs are capable of so much logging, that they can quickly overwhelm the routing engine if security logging is attempted via the control plane (out the fxp0 interface).

To overcome this important aspect of logging security events, an administrator can dedicate a revenue port for logging tasks. Doing so will cause logging for security events to be sent out the SRX from the data plane, rather than the control plane, resembling the behavior of the branch SRX devices that don't have a dedicated hardware control plane.

The logging parameters configured in the security policies are unchanged regardless of the way you do the logging, providing you learn how to configure using these three methodologies:

- Logging via a revenue port (applicable to SRX branch and high-end).
- Logging via the control plane (applicable to SRX high-end).
- Logging to a NSM server (applicable to SRX branch and high-end).

To Configure Logging Via a Revenue Port (data plane):

1. Configure the logging mode and format. Typical formats used are `syslog` (standard) and `sd-syslog` (structured):

```
[edit]
root# set security log mode stream

[edit]
root# set security log format sd-syslog
```

2. Configure the source of messages and the syslog server. A name is required for the server, but this is locally significant, and you are encouraged to use something meaningful to facilitate its identification when troubleshooting or auditing the firewall config:

```
[edit]
root# set security log source-address 192.168.2.1

[edit]
root# set security log stream SYSLOG_SERVER host 192.168.2.2
```

NOTE Other options, such as the destination port, can be configured in case you are not using the default (UDP port 1514). Also note that the host 192.168.2.2, does not have any syslog services, but it is being configured here for example purposes.

To Configure Logging Via the Control Plane (out fxp0):

1. Configure the logging format. Options available are `syslog` (standard) and `sd-syslog` (structured):

```
[edit]
root# set security log format sd-syslog
```

2. Configure the source of messages and the syslog server. Once again, a name is required for the server, but it is locally significant and you are encouraged to use something meaningful to facilitate its identification when troubleshooting or auditing the firewall config:

```
[edit]
root# set security log source-address 10.189.140.99

[edit]
root# set security log stream SYSLOG_SERVER host 10.189.132.70
```

3. Configure logging via the control plane, so any logs generated in the data plane as a consequence of a security policy match, are sent to the control plane:

```
[edit]
root# set security log mode event
```

4. Configure rate limiting of event logs to the control plane. A good practice is to allow no more than 1,000 log entries per second:

```
[edit]
root# set security log mode event event-rate 1000
```

To Configure Logging to NSM:

NOTE By default, SRX devices do not send native syslog messages to NSM, only the logs stored in two files in the SRX. If logging is from a high-end SRX, then security logs must be sent to the control plane first.

1. Configure the logging mode and format. Options available for the format are `syslog` (standard) and `sd-syslog` (structured):

```
[edit]
root# set security log mode event

[edit]
root# set security log format sd-syslog
```

2. Configure the source of traffic and NSM as the target syslog server. A name is required for the target, but it is locally significant, and you are encouraged to use something to facilitate its identification when troubleshooting or auditing the firewall config:

```
[edit]
root# set security log source-address 10.189.140.99

[edit]
root# set security log stream SYSLOG_SERVER host 10.189.132.72
```

3. Configure logging via the control plane and rate limit the entries (this step is only required for high-end SRX, so omit if you are configuring a branch device):

```
[edit]
root# set security log mode event event-rate 1000
```

4. Configure logging to NSM. The following entries are mandatory for properly logging system and security logs:

```
[edit]
root# set system syslog file default-log-messages structured-data

[edit]
root# set system syslog file default-log-messages any any
```

That's it. You should have your first security policies running and a method to log them. Let's move on now to learning how to prepare those packets that are leaving the policy in route for their destinations.

Chapter 7

Configuring NAT Source

By default, incoming packets to the SRX are routed to the destination. You can easily confirm this by inspecting the session table. For example, the (truncated) results of the `show security flow session` command from our test performed earlier, revealed the following for one of the flows:

```
[edit security policies]
root# run show security flow session
Session ID: 100001790, Policy name: admins_to_untrust/4, Timeout: 1800
  In: 192.168.2.2/4781 --> 209.239.112.126/80;tcp, If: ge-0/0/0.0
  Out: 209.239.112.126/80 --> 192.168.2.2/4781;tcp, If: ge-0/0/2.0

<snip>
```

The output indicates that there is an incoming packet with a source IP address of 192.168.2.2, destined to 209.239.112.126. The return traffic shows a packet being sourced from 209.239.112.126, destined to the exact IP address that originated the transaction, 192.168.2.2.

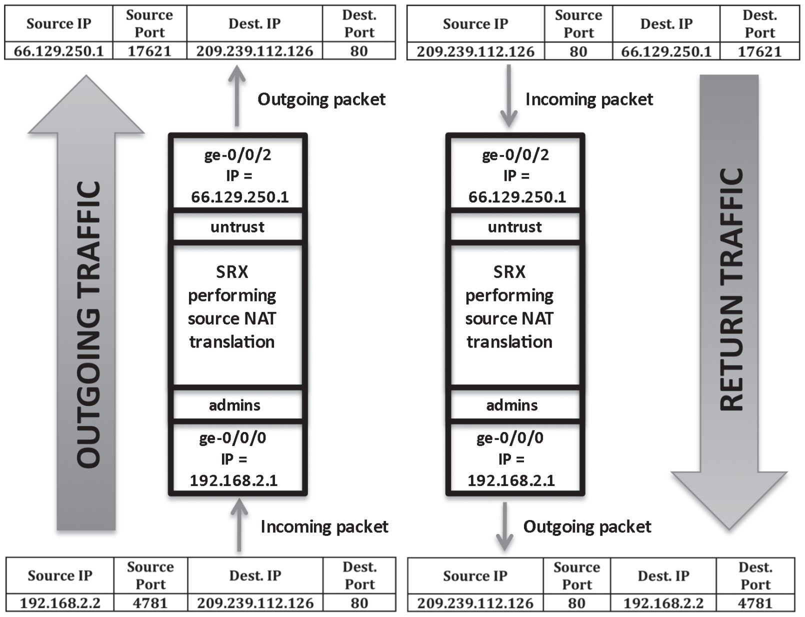
When the source and destination IP addresses remain unchanged, as in this example, it is indicative that the traffic was routed as opposed to translated (NATed).

NAT Types

The SRX is capable of performing different forms of translation of the source and destination headers. The options are: source, destination, and static.

This chapter shows you how to configure source NAT to translate the source IP of incoming packets as they leave the SRX. There are several options available when executing this kind of translation. For example, you can configure it so that the source IP is translated to the IP of the egress interface, you can use a different pool of IP addresses and no port address translation, and there are also a few more options.

Figure 7.1 illustrates an example of source NAT. Aside from translating the source IP to that of the interface ge-0/0/2, the SRX also translates the source port (performs port address translation or PAT by default). Translating the source IP allows sharing a single IP address between thousands of hosts, while translating the source port gives the SRX the possibility of tracking who initiated a particular flow, and the ability to hand off the return traffic to the corresponding host that opened the connection.



Source IP	Source Port	Dest. IP	Dest. Port
66.129.250.1	17621	209.239.112.126	80

Source IP	Source Port	Dest. IP	Dest. Port
209.239.112.126	80	66.129.250.1	17621

Source IP	Source Port	Dest. IP	Dest. Port
192.168.2.2	4781	209.239.112.126	80

Source IP	Source Port	Dest. IP	Dest. Port
209.239.112.126	80	192.168.2.2	4781

OUTGOING TRAFFIC
Outgoing packet
ge-0/0/2 IP = 66.129.250.1
untrust
SRX performing source NAT translation
admins
ge-0/0/0 IP = 192.168.2.1
Incoming packet
Incoming packet
ge-0/0/2 IP = 66.129.250.1
untrust
SRX performing source NAT translation
admins
ge-0/0/0 IP = 192.168.2.1
Outgoing packet
RETURN TRAFFIC

Figure 7.1 Source NAT Using Egress Interface

Source NAT is not configured to enable Internet access to many hosts. In fact, any of the translation types offered in the SRX have no concept of "Internet" or "trusted segment." The SRX focuses on zones, and from that perspective when you configure different NAT services think about the context of the traffic (the from-zone and to-zone). Once you have defined the context, then decide what is it that you want to do, is it translation of the source IPs? Is it translation of the destination IPs? Maybe both? The answer to these questions helps you decide on the type of NAT that you need to configure.

Before jumping onto the console again, it is important to know that NAT is decoupled from the security policies. This means that the configuration is done in a different hierarchy, under `[edit security nat]`. This offers an incredible level of flexibility to the administrators by permitting reconfiguration of Layer 3 operations without any effect on security policies.

WARNING Do not be confused here. Remember that in order for traffic to go across the SRX you need to configure a security policy. A source NAT configuration looks very similar to a security policy, but this will not allow the traffic through, it will only manipulate the traffic once it has been permitted by the security policy.

Let’s configure the following source NAT scenarios:

- Using the IP address of the egress interface.
- Using a translation pool.
- Creating a rule to except traffic.

As you commit the different scenarios make use of show commands to verify their correctness and operation. To verify correctness you can use the following command and any of its available parameters:

```
root# run show security nat source ?
Possible completions:
  persistent-nat-table  Show persistent NAT table information
  pool                Show source NAT information of this pool
  rule                Show source NAT rule-set information
  summary              Show source NAT summary information
```

The actual operation can be demonstrated with the `show security flow session` command as used previously in this book. If you decide to use this command, then pay close attention to how the destination IP address for the response changes. Remember that this command offers the following multiple parameters that you can use to filter the output:

```
root# run show security flow session ?
Possible completions:
  <[Enter]> Execute this command
  application Show session for specified application or application set
  destination-port  Show each session that uses specified destination port
  destination-prefix  Show each session that matches destination prefix
  idp     Show IDP sessions
  interface   Show each session that uses specified interface
  protocol  Show each session that uses specified IP protocol
  resource-manager Show resource-manager sessions
  session-identifier  Show session with specified session identifier
  source-port   Show each session that uses specified source port
  source-prefix Show each session that matches source prefix
  summary   Show summary of sessions
  tunnel  Show tunnel sessions
  |       Pipe through a command
```

Configuring Source NAT Using the Egress Interface

You are going to configure source NAT to permit hosts in the "admins" zone Internet access, translating the source IP address of the incoming packets to the IP address of the egress interface in the "untrust" zone (ge-0/0/2).

Please refer to Figure 2.2 , our book's network topology, for a full understanding of the following example.

To Configure Source NAT Using the Egress Interface:

1. Create a NAT source rule-set. Give this a meaningful name that describes what the rule-set will do:

```
[edit]
root# edit security nat source rule-set internet_nat
```

2. Define the context of the traffic. Where is it coming from and where is going to?

```
[edit security nat source rule-set internet_nat]
root# set from zone admins

[edit security nat source rule-set internet_nat]
root# set to zone untrust
```

3. Now that you have created a rule-set, and defined the context of the traffic, configure an actual rule that matches all the incoming traffic from the "admins" subnet going to any location, and NAT source that using the egress interface. Again, choose a name for the rule that is meaningful to you:

```
[edit security nat source rule-set internet_nat]
root# edit rule admins_access

[edit security nat source rule-set internet_nat rule admins_access]
root# set match source-address 192.168.2.0/24

[edit security nat source rule-set internet_nat rule admins_access]
root# set match destination-address all

[edit security nat source rule-set internet_nat rule admins_access]
root# set then source-nat interface

[edit security nat source rule-set internet_nat rule admins_access]
root# commit
commit complete
```

Before continuing, take a moment to analyze what you just configured.

Think of a rule-set as a container where you can add individual rules. This rule-set has a context, in this case you specified zones for the context (from zone "admins" to zone "untrust"), but you could also use interfaces or virtual routers.

The rules are the individual statements that we are looking for, and if found, then a specific action is taken. A no match in any rule means that the SRX will not modify the headers, just as when packets are being routed.

This IF-THEN-ELSE algorithm is similar to the configuration of security policies. As you may suspect, rules within a rule-set are processed in a top-down fashion, so their relative order is important. Make sure that you place the most specific rules on top. The power behind this concept is revealed later in this chapter.

4. At this time, you might initiate a few connections to the Internet from the PC in the "admins" zone, and then inspect the session table for results:

```
root> show security flow session
Session ID: 100024536, Policy name: admins_to_untrust/4, Timeout: 1772
  In: 192.168.2.2/3520 --> 64.94.18.157/443;tcp, If: ge-0/0/0.0
  Out: 64.94.18.157/443 --> 66.129.250.1/64507;tcp, If: ge-0/0/2.0

Session ID: 100024537, Policy name: admins_to_untrust/4, Timeout: 1790
  In: 192.168.2.2/3525 --> 77.223.130.61/5938;tcp, If: ge-0/0/0.0
  Out: 77.223.130.61/5938 --> 66.129.250.1/64509;tcp, If: ge-0/0/2.0

Session ID: 100024549, Policy name: admins_to_untrust/4, Timeout: 32
  In: 192.168.2.2/47621 --> 66.56.36.86/27374;udp, If: ge-0/0/0.0
  Out: 66.56.36.86/27374 --> 66.129.250.1/64507;udp, If: ge-0/0/2.0

Session ID: 100024550, Policy name: admins_to_untrust/4, Timeout: 32
  In: 192.168.2.2/47621 --> 66.57.86.100/40644;udp, If: ge-0/0/0.0
  Out: 66.57.86.100/40644 --> 66.129.250.1/41683;udp, If: ge-0/0/2.0

<snip>
```

Notice the "Out" line for every session – the responses are being directed to the IP address 66.129.250.1. This IP address corresponds to the SRX egress interface, confirming the proper operation of source NAT.

Configuring Source NAT Using Translation Pools

Depending on the size of the organization, one IP address and port address translation (PAT) might be insufficient, so a pool of IP addresses is needed to accommodate this demand.

Another scenario may require the translation of an entire subnet to a completely different addressing scheme. This application is common in environments that require connectivity between overlapping networks, like in company mergers.

Let's reconfigure the previous example to perform the translation using a pool of IP addresses. Once again, refer to Figure 2.2, this book's network topology, for a full understanding of the example.

To Configure Source NAT Using a Translation Pool:

1. Create a pool of addresses (66.129.250.10 - 66.129.250.15) that will be used as the source IP for the outgoing packets. Give this pool a meaningful name, describing its purpose for future reference:

```
[edit security nat source]
root# set pool public_NAT_range address 66.129.250.10 to 66.129.250.15
```

2. Apply the pool just created. Pools can be reused across multiple rules, so think exactly about this application – here you want to modify the rule `admin_access` under the `internet_nat` rule-set:

```
[edit security nat source]
root# edit rule-set internet_nat rule admins_access

[edit security nat source rule-set internet_nat rule admins_access]
root# set then source-nat pool public_NAT_range
```

3. Verify that the configuration looks similar to this:

```
[edit security nat source rule-set internet_nat rule admins_access]
root# show
match {
   source-address 192.168.2.0/24;
   destination-address 0.0.0.0/0;
}
then {
   source-nat {
      pool {
         public_NAT_range;
      }
   }
}
```

4. Commit and confirm the operation:

```
 [edit security nat source rule-set internet_nat rule admins_access]
root# commit
commit complete
```

The result of this configuration should be that outgoing packets for any host in the "admins" zone are translated randomly to one of the IP addresses in the `public_NAT_range`.

To illustrate the flexibility and power of rules and rule-sets, let's now configure Scenario 3 from the beginning of this chapter: Create a rule to except traffic.

And here it's necessary to create a rule so that traffic initiated by a host (192.168.2.2) is not translated as per the `admin_access` rule.

To Configure a Source NAT Exception Rule:

1. Create a new rule under the internet_nat rule-set. Give the new rule a descriptive name:

```
[edit security nat source rule-set internet_nat rule admins_access]
root# up
[edit security nat source rule-set internet_nat]
root# set rule NO_translate

[edit security nat source rule-set internet_nat]
root# edit rule NO_translate
```

2. Define the match criteria (what you are going to except from translation):

```
[edit security nat source rule-set internet_nat rule NO_translate]
root# set match source-address 192.168.2.2/32

[edit security nat source rule-set internet_nat rule NO_translate]
root# set match destination-address all
```

3. Configure the action when that particular source/destination is identified:

```
[edit security nat source rule-set internet_nat rule NO_translate]
root# set then source-nat off
```

4. Reorganize the rules so that NO_translate is evaluated first:

```
[edit security nat source rule-set internet_nat rule NO_translate]
root# up
```

```
[edit security nat source rule-set internet_nat]
root# insert rule NO_translate before rule admins_access
```

IMPORTANT This step is crucial, and if you forget about it or ignore it, then the host 192.168.2.2 will continue to be translated. Remember that rules in NAT rule-sets are evaluated in a top-down fashion like a security policy, so there is always a need to analyze and reorganize the rules when necessary.

5. Verify the configuration. This should look similar to the following:

```
[edit security nat source rule-set internet_nat]
root# show
from zone admins;
to zone untrust;
rule NO_translate {
   match {
      source-address 192.168.2.2/32;
      destination-address 0.0.0.0/0;
   }
   then {
      source-nat {
         off;
      }
   }
}
rule admins_access {
   match {
      source-address 192.168.2.0/24;
      destination-address 0.0.0.0/0;
   }
   then {
      source-nat {
         interface;
      }
```

There. We've accomplished our three items for NAT. Our SRX is properly analyzing and sending traffic according to our network topology as shown way back in Chapter 2.

After completing the initial configuration you can continue the configuration of more services supported by the SRX. Some of these services are: IPS, IPSec, UTM, high-availability, and more advanced routing configurations. The online documentation available provides a lot of details on how to configure every service, and this documentation gets updated with every new Junos release. Look for it here: http://www.juniper.net/techpubs/hardware/junos-srx/index.html.

More Day One books are also being developed, especially for the SRX platform of services gateway. Keep checking the website for new additions at www.juniper.net/daysone.

And finally, if you need more help, get the *Junos Security* book from O'Reilly publishers. Look for it here: www.juniper.net/books.

Now let's apply what we've done to multiple SRX devices as efficiently as possible.

Chapter 8

Importing the SRX into NSM

If you are installing and managing multiple SRX devices and other Juniper hardware, the Network and Security Manager (NSM) will help you keep a more consistent view of the network, and will simplify your configuration and troubleshooting tasks. In this chapter you'll learn how to import your SRX into NSM. The installation and configuration of NSM, and the devices already imported, are out of the scope of this book. The intention here is to jump start NSM usage by connecting your SRX and NSM, and to ensure that the SRX is properly imported so that you can continue managing it via NSM.

Preparing the SRX for NSM Connectivity

Preparing the SRX for NSM connectivity is a direct process. In fact, if you have been closely following the sequence of events in this book, you have almost completed all the steps required.

To recap, you need to configure:

- A user account with administrator privileges
- IP addressing and routing reachability
- SSH access
- Netconf access

You already have the first three components, so you must configure netconf support. Note, however, if you are using a revenue port to connect to NSM, then more work is necessary, namely the upfront set up of zones and the host-inbound-traffic.

The work in this chapter is being done on the SRX3400. As throughout this book, please refer to Figure 2.2 for details of the network topology.

To Configure netconf Support:

First, enable netconf support:

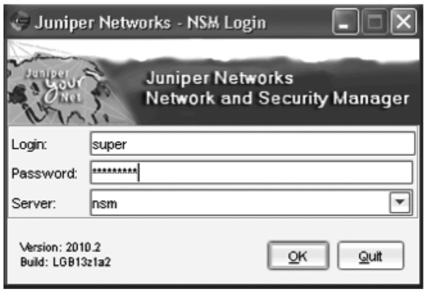
```
[edit]
root# set system services netconf ssh

[edit]
root# commit
commit complete
```

Importing the SRX into NSM

Now we'll log into the Network and Security Manager server, and import the SRX. The NSM is a graphical tool, and the steps required to import the SRX are presented in screen captures instead of CLI snippets.

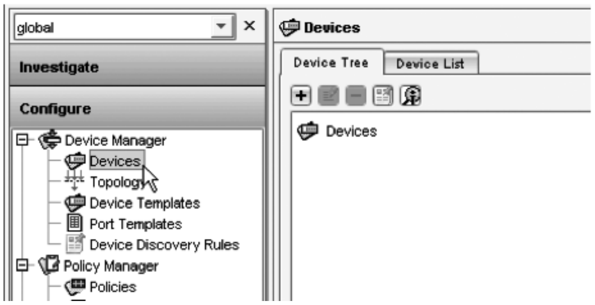
To Import SRX into NSM:

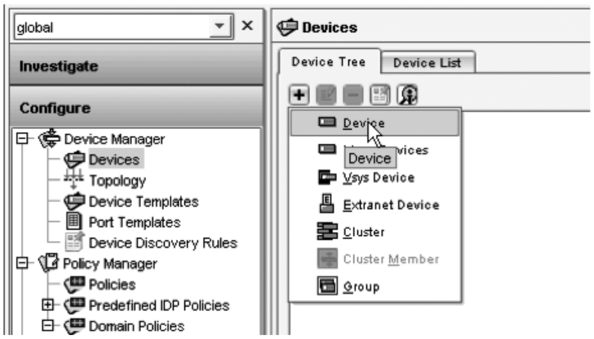
1. Log in to the NSM server as root, or an administrator, with permissions to import the SRX3400.



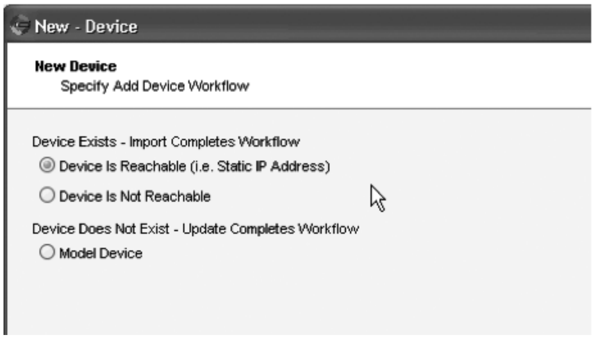
Login: super
Password: *********
Server: nsm
Version: 2010.2
Build: LGB13z1a2

2. Under devices click the "+" to add a new device.





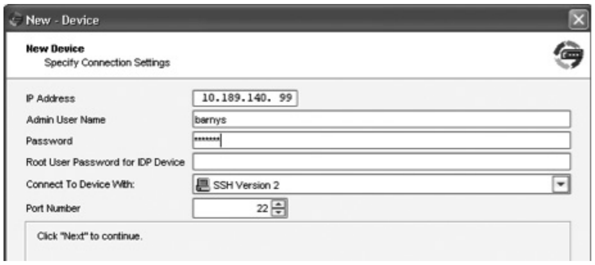
3. Specify that the device is reachable and fill in the IP address and administrator account fields. The IP address in this example is that of the fxp0 interface, and the administrator account is from the SRX, not the NSM.



New - Device

New Device
Specify Add Device Workflow

Device Exists - Import Completes Workflow
[✓ Device Is Reachable (i.e. Static IP Address)]
Device Is Not Reachable
Device Does Not Exist - Update Completes Workflow
Model Device



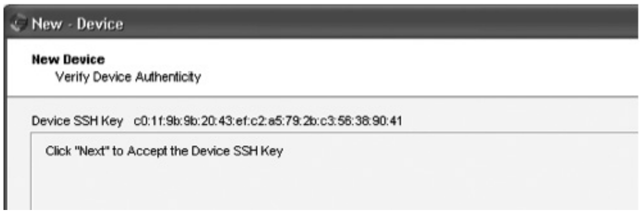
New - Device

New Device
Specify Connection Settings

IP Address	10.189.140. 99
Admin User Name	barnys
Password	*******
Root User Password for IDP Device	
Connect To Device With:	SSH Version 2
Port Number	22

Click "Next" to continue.

4. Click “Next” to accept the SSH keys.



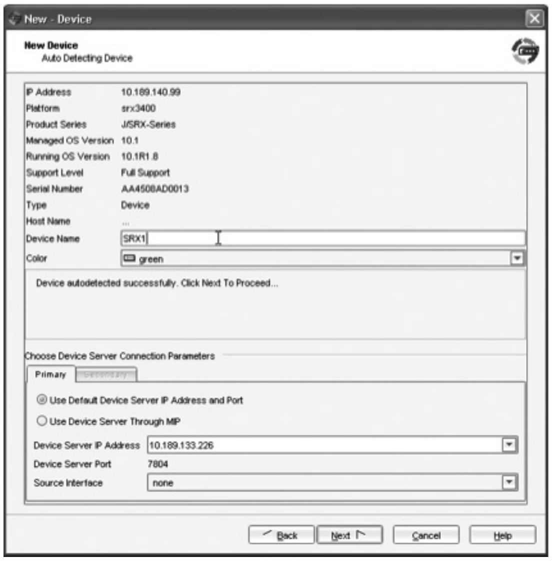
New - Device

New Device
Verify Device Authenticity

Device SSH Key c0:1f:9b:9b:20:43:ef:c2:a5:79:2b:c3:56:38:90:41

Click "Next" to Accept the Device SSH Key

5. When the device is auto detected, click “Next” to confirm to NSM that you wish to import it. If you did not configure a hostname for the device during initial installation, then you can specify one here before importing the device, otherwise the “Device Name” will be already populated.



New Device
Auto Detecting Device

IP Address	10.189.140.99
Platform	srx3400
Product Series	J/SRX-Series
Managed OS Version	10.1
Running OS Version	10.1R1.8
Support Level	Full Support
Serial Number	AA4508AD0013
Type	Device
Host Name	...
Device Name	SRX1
Color	green

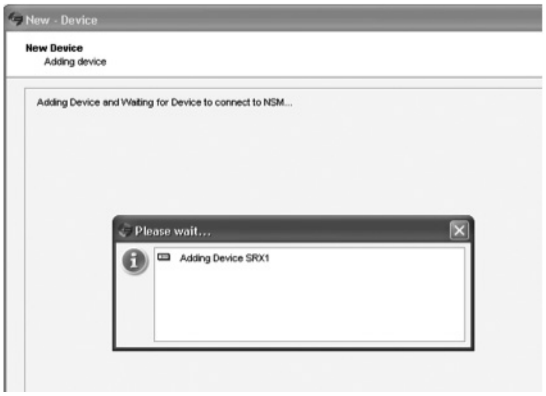
Device autodetected successfully. Click Next To Proceed...

Choose Device Server Connection Parameters

Primary

(●) Use Default Device Server IP Address and Port
( ) Use Device Server Through MIP

Device Server IP Address	10.189.133.226
Device Server Port	7804
Source Interface	none

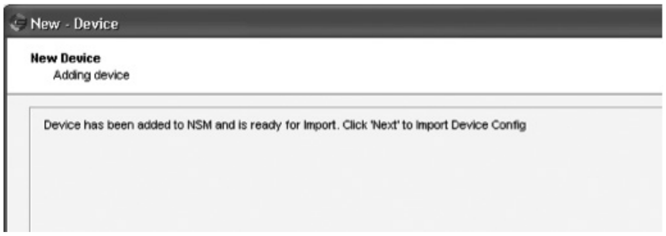


New Device
Adding device

Adding Device and Waiting for Device to connect to NSM...

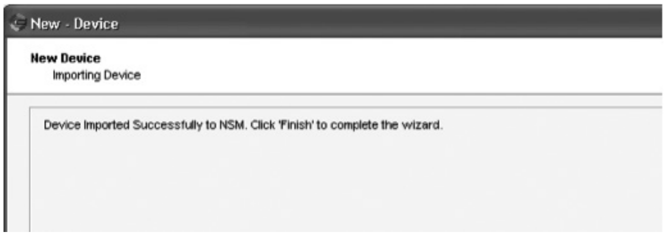
Adding Device SRX1

6. Import the configuration of SRX into NSM.



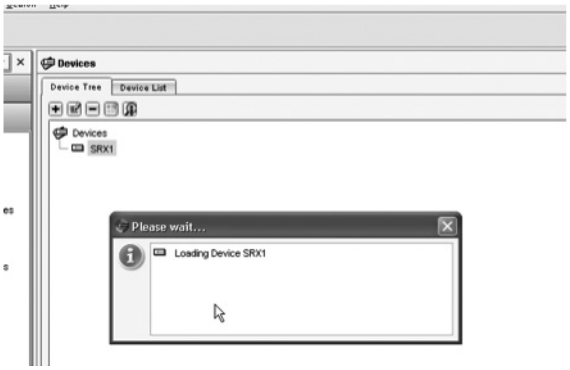
New Device
Adding device

Device has been added to NSM and is ready for import. Click 'Next' to Import Device Config



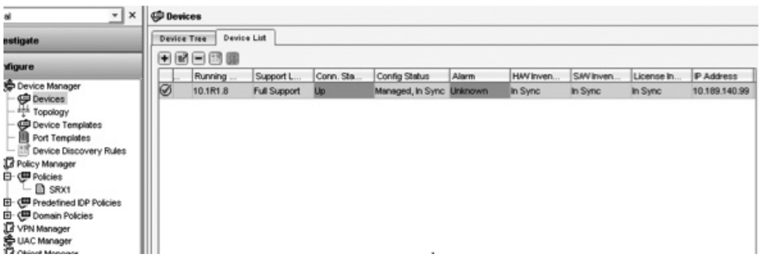
New Device
Importing Device

Device Imported Successfully to NSM. Click 'Finish' to complete the wizard.



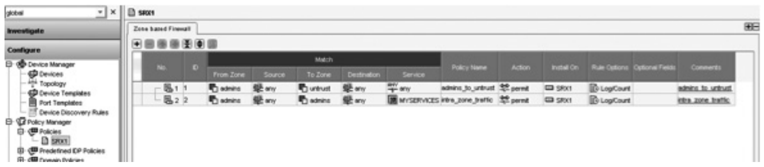
Loading Device SRX1

7. Confirm that the device is properly connected and in sync with NSM.



	...	Running ...	Support L...	Conn. Sta...	Config Status	Alarm	H/W Inven...	S/W Inven...	License In...	IP Address
✓		10.1R1.8	Full Support	Up	Managed, In Sync	Unknown	In Sync	In Sync	In Sync	10.189.140.99

8. Inspect the device configuration in NSM, such as the security policies. Proceed with any management functions from here.



No.	ID	Match					Policy Name	Action	Install On	Rule Options	Optional Fields	Comments
		From Zone	Source	To Zone	Destination	Service						
1	1	admins	any	untrust	any	any	admins_to_untrust	permit	SRX1	Log/Count		admins_to_untrust
2	2	admins	any	admins	any	MYSERVICES	intra_zone_traffic	permit	SRX1	Log/Count		intra_zone_traffic

From this moment on, you can continue managing the SRX via NSM. Any changes done to the SRX via J-Web, or the CLI, are overwritten by NSM in the next configuration update, unless you import the device configuration first.

Chapter 9

Troubleshooting Tools

When things go wrong it is good to have a survival kit to help to know what to do. Hence this chapter provides a few tips that serve as a starting point to detect and fix common network connectivity problems.

The SRX offers many different services, and this book explores only a few of them. Writing a comprehensive guide of all the SRX services and how to troubleshoot these warrants many Day One books and the development of more advanced literature.

MORE? *Junos Security*, recently published by O'Reilly Media, has several troubleshooting sections and case study scenarios that can be helpful to readers of this book. For more info, see www.junper.net/books.

Understanding Flow Processing

As you configure more services in the SRX, you need to know exactly how packets are processed in the device. The series of steps the SRX takes each time a packet enters an interface up through when it exits the device is known as *flow processing*. Understanding the packet processing gives you insight into what the SRX processes first and the relative order of services. And knowing this process can help you to better pinpoint potential failures, making you more efficient in your diagnostics.

Figure 9.1 shows where the flow processing takes place during the packet stay within the device.

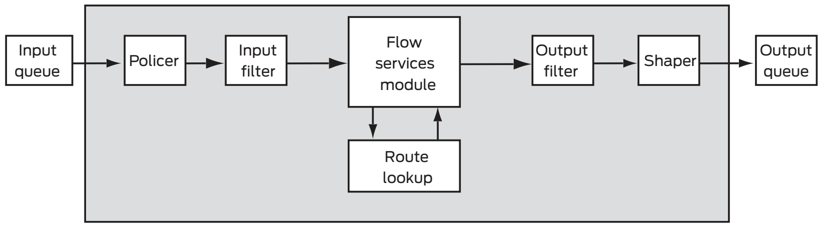


Figure 9.1 **Flow Processing**

Now examine what is done inside the flow services module represented in Figure 9.2. Screens, NAT, ALG, and IPSec, are only some of the services performed by this module. Figure 9.1 and 9.2 are explained in great detail in the *Junos Security Configuration Guide* found here (for version 10.1): https://www.juniper.net/techpubs/software/junos-srx/junos-srx10.1/index.html.

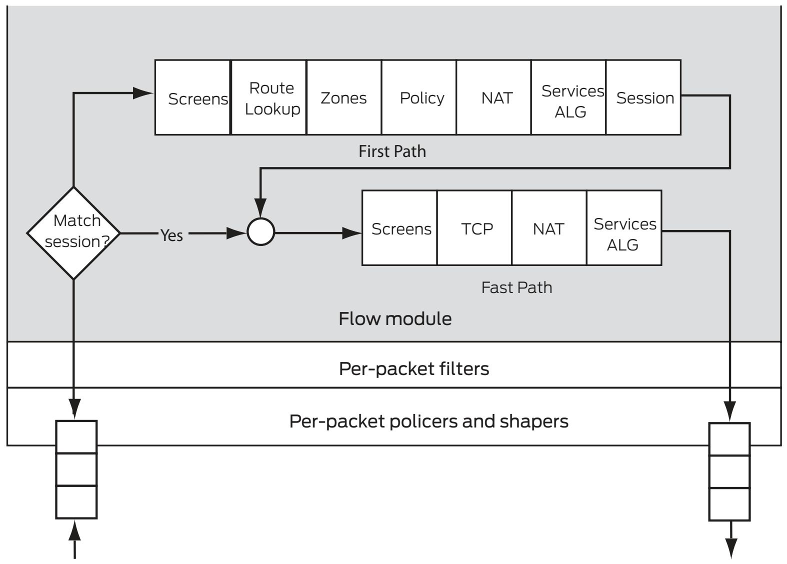


Figure 9.2 **Flow Services Module**

As you spend more time in the SRX and uncover its capabilities for more advanced services, you will realize how important it is to know this information. So when time permits, spend a few minutes in the aforementioned guide reading the more detailed explanations of how the packets are processed – it is time well invested if you want to become proficient with administering SRX devices.

Additionally, it's important to mention that branch SRX devices are capable of bypassing the flow services module for selected configurations. This is called packet-based mode processing (as opposed to flow-based processing). The extra flexibility of these platforms makes it a little more difficult to fully understand the packet processing through the SRX.

Examining Logs and System Status

Logging was configured in Chapter 5. Return to it to refresh your mind, because we're about to re-stress the flexibility of this tool.

Logging to a file can be done with the `show log [log_file]` command.

You can log events of many different kinds, as shown throughout this chapter, however, you will also find log files that were not configured by you or that exist by default. One such log file is *messages*, used by the system to log kernel and services messages.

To look for failed password attempts from admins attempting to connect via SSH, you can try the following combination:

```
> show log messages | match ssh | match “Failed password” | trim 38
```

This command would look in the `messages` log file, and match for sshd events where “Failed password” attempts were recorded. The “trim 38” simply omits the first 38 characters from the output if you are not interested in the time stamps.

Another outstanding option when dealing with logs is to use the `start monitor` command, which allows you to monitor a log file in real time, so when an event happens it is displayed in console right away. Here is an example where you look for failed ssh attempts in real time.

To Use the Start Monitor Tool :

1. Start monitoring the log file (and apply match statements if you want to narrow your search):

```
barnys@SRX3400> monitor start messages | match “ssh” | match “Failed password”
```

2. Press “Esc-Q” to enable and disable the output display to console as needed:

```
barnys@SRX3400>
*** monitor and syslog output enabled, press ESC-Q to disable ***

*** messages ***
Jun 27 00:19:54  SRX3400 sshd[64008]: Failed password for john from 10.188.133.42 port
50021 ssh2

barnys@SRX3400>
*** monitor and syslog output disabled, press ESC-Q to enable ***
```

3. Stop the real time monitoring of the file. This does not cause logging to stop recording events, but the events are not shown on the console anymore:

```
barnys@SRX3400> monitor stop
```

Try It Yourself: Examing the System Status

Examining the system status can be done with show commands. You have used several show commands if you've been following the examples in this book, but many more options are also available to find out chassis and event status information. Try these on your console right now.

To check system uptime:

```
> show system uptime
```

To check the software version:

```
> show version
```

To check for active alarms:

```
> show system alarms
> show chassis alarms
```

There are many parameters available to check chassis related information, so check the help prompt to see them all:

```
> show chassis ?
```

To find out the running processes and CPU utilization:

```
> show system processes extensive
```

MORE? The complete list of available show commands and their descriptions can be found in the *CLI Reference Guide*, found here (for version 10.1) at www.juniper.net/techpubs/software/junos-srx/junos-srx10.1/index.html. You can also find lots of device agnostic command usage examples in Day One books from the *Junos Fundamentals Series*: www.juniper.net/dayone.

Enabling Traceoptions

Traceoptions are the equivalent of debugging tools from other vendors' products, so if you are coming from a background of using Juniper Networks ScreenOS firewalls and are familiar with `debug flow basic`, then this section will show you how to do the same in the SRX.

To debug packets as they traverse the SRX, you need to configure traceoptions and flag basic-datapath. This will trace packets as they enter the SRX until they exit, giving you details of the different actions the SRX is taking along the way.

To use traceoptions flag basic-datapath :

1. Enable `traceoptions flag basic-datapath`. Capture the results to a file of your preference:

```
barnys@SRX3400# set security flow traceoptions file DEBUG
barnys@SRX3400# set security flow traceoptions flag basic-datapath
```

2. Configure a packet-filter to match traffic going one way (outbound in this case):

```
barnys@SRX3400# set security flow traceoptions packet-filter match-outgoing source-
prefix 192.168.2.0/24
barnys@SRX3400# set security flow traceoptions packet-filter match-outgoing destination-
prefix 0.0.0.0/0
```

3. Configure a packet-filter to match the reverse or response traffic:

```
barnys@SRX3400# set security flow traceoptions packet-filter match-reverse source-prefix
0.0.0.0/0
barnys@SRX3400# set security flow traceoptions packet-filter match-reverse destination-
prefix 192.168.2.0/24
```

NOTE Steps 2 and 3 are necessary because Junos only captures one directional flows. Multiple packet-filters then let you capture both the outgoing and reverse flows. Individual packet-filter configurations like in this example are processed as OR statements, instructing the SRX to match traffic that matches one filter *or* the other.

4. After committing this configuration, you can inspect the capture results with the `show log DEBUG` command.

MORE? Troubleshooting is a huge topic whose surface has barely been scratched here, but as with everything in this short book, it's a jump start to a much larger SRX world. Use other Day One books to assist you in troubleshooting, the new SRX book, *Junos Security* from O'Reilly Media, and of course, the SRX documentation.

Appendix

SRX Default Factory Configurations

Here are several SRX default factory configurations for your reference.

SRX210 Default Factory Configuration. Loaded Junos 10.1R18.

```
## Last commit: 2010-03-23 08:39:12 UTC by root
version 10.1R1.8;
system {
    autoinstallation {
        delete-upon-commit; ## Deletes [system autoinstallation]
upon change/commit
        traceoptions {
            level verbose;
            flag {
                all;
            }
        }
        interfaces {
            ge-0/0/0 {
                bootp;
            }
        }
    }
    name-server {
        208.67.222.222;
        208.67.220.220;
    }
    services {
        ssh;
        telnet;
        web-management {
            http {
                interface vlan.0;
            }
            https {
                system-generated-certificate;
                interface vlan.0;
            }
        }
        dhcp {
            router {
                192.168.1.1;
            }
            pool 192.168.1.0/24 {
                address-range low 192.168.1.2 high 192.168.1.254;
            }
            propagate-settings ge-0/0/0.0;
        }
    }
    syslog {
        archive size 100k files 3;
        user * {
            any emergency;
        }
        file messages {
            any critical;
            authorization info;
        }
        file interactive-commands {
            interactive-commands error;
        }
    }
    max-configurations-on-flash 5;
    max-configuration-rollbacks 5;
    license {
        autoupdate {
            url https://ae1.juniper.net/junos/key_retrieval;
        }
    }
    ## Warning: missing mandatory statement(s): 'root-
authentication'
}
interfaces {
    interface-range interfaces-trust {
        member ge-0/0/1;
        member fe-0/0/2;
        member fe-0/0/3;
        member fe-0/0/4;
        member fe-0/0/5;
        member fe-0/0/6;
        member fe-0/0/7;
        unit 0 {
            family ethernet-switching {
                vlan {
                    members vlan-trust;
                }
            }
        }
    }
    ge-0/0/0 {
        unit 0;
    }
    vlan {
        unit 0 {
            family inet {
                address 192.168.1.1/24;
            }
        }
    }
}
security {
    nat {
        source {
            rule-set trust-to-untrust {
                from zone trust;
                to zone untrust;
                rule source-nat-rule {
                    match {
                        source-address 0.0.0.0/0;
                    }
                    then {
                        source-nat {
                            interface;
                        }
                    }
                }
            }
        }
    }
    screen {
        ids-option untrust-screen {
            icmp {
                ping-death;
            }
            ip {
                source-route-option;
                tear-drop;
            }
            tcp {
                syn-flood {
                    alarm-threshold 1024;
                    attack-threshold 200;
                    source-threshold 1024;
                    destination-threshold 2048;
                    timeout 20;
                }
                land;
            }
        }
    }
    zones {
        security-zone trust {
            host-inbound-traffic {
                system-services {
                    all;
                }
                protocols {
                    all;
                }
```

```
         }
         interfaces {
            vlan.0;
         }
      }
      security-zone untrust {
         screen untrust-screen;
         interfaces {
            ge-0/0/0.0 {
               host-inbound-traffic {
                  system-services {
                     dhcp;
                     tftp;
                  }
               }
            }
         }
      }
   }
   policies {
      from-zone trust to-zone untrust {
         policy trust-to-untrust {
            match {
               source-address any;
               destination-address any;
               application any;
            }
            then {
               permit;
            }
         }
      }
   }
}
wlan {
   cluster vlan-0-default {
      name juniper-ap-cluster;
      default-cluster;
      interfaces {
         vlan.0;
      }
   }
}
vlans {
   vlan-trust {
      vlan-id 3;
      l3-interface vlan.0;
   }
}
```

SRX3400 Default Factory Configuration. Loaded Junos 10.1R18.

```
## Last commit: 2010-03-24 14:35:41 UTC by root
version 10.1R1.8;
system {
    syslog {
        user * {
            any emergency;
        }
        file messages {
            any notice;
            authorization info;
        }
        file interactive-commands {
            interactive-commands any;
        }
    }
    license {
        autoupdate {
            url https://ae1.juniper.net/junos/key_retrieval;
        }
    }
    ## Warning: missing mandatory statement(s): 'root-
authentication'
}
security {
    idp {
        security-package {
            url https://services.netscreen.com/cgi-bin/index.cgi;
        }
    }
}
```

Reviewing and Applying Licenses

You can check the licenses needed in your SRX and apply them following a few simple steps.

Licensing is usually ordered when the device is purchased, and this is bound to the chassis serial number. These instructions assume that you already have the license in your possession, but if that is not the case, then work with your account team, or place a call to Juniper's customer care, to get the assistance that you need.

If you have a letter and all you need is to generate the licenses, or if you need the contact information for customer care, please visit https://www.juniper.net/generate_license/.

The steps below demonstrate the actual process of checking and applying a license.

To Apply a License:

1. Connect to your device and check the status of your license:

```
root> show system license
License usage: none
Licenses installed: none
```

2. The output in this case indicates that there are no features configured that require licensing. An example would be if IPS services were configured.

Double-check that the license received matches your chassis's serial number. To view the serial number:

```
root> show chassis hardware
Hardware inventory:
Item          Version  Part number  Serial number
Description
Chassis                             AA4508AD0013     SRX 3400
…
```

3. Open the license file received in notepad or any other text editor. Copy all of its contents to the clipboard by highlighting everything, and then clicking Edit->Copy.

4. On the SRX, type the following operational mode command, and copy the contents of the clipboard, followed by Ctrl-D (escape sequence):

```
root> request system license add terminal
[Type ^D at a new line to end input,
 enter blank line between each license key]
JUNOS252544 aeaqea qmifat injqhb auimbq gezqqb qcdw4x
           loxva4 ueffko xw4whk 6mfqs6 7oyg5t zn2j5k
           upc2ff bxrvkt 4ut24u w44joc n24g4x 7pevbz
           psq
JUNOS252544: successfully added
add license complete (no errors)

root>
```

5. Although the output indicates no errors, it is a good practice to confirm that the license was applied properly:

```
root> show system license
License usage:
                                 Licenses     Licenses    Licenses    Expiry
  Feature name                       used    installed      needed
  idp-sig                               0            1           0      2011-04-29 00:00:00 UTC

Licenses installed:
  License identifier: JUNOS252544
  License version: 2
  Valid for device: AA4508AD0013
  Features:
    idp-sig          - IDP Signature
      date-based, 2010-04-10 00:00:00 UTC - 2011-04-29 00:00:00 UTC

root>
```

Licenses take effect as soon as you apply them, so there is no need to reboot your device.

Steel-Belted RADIUS Integration

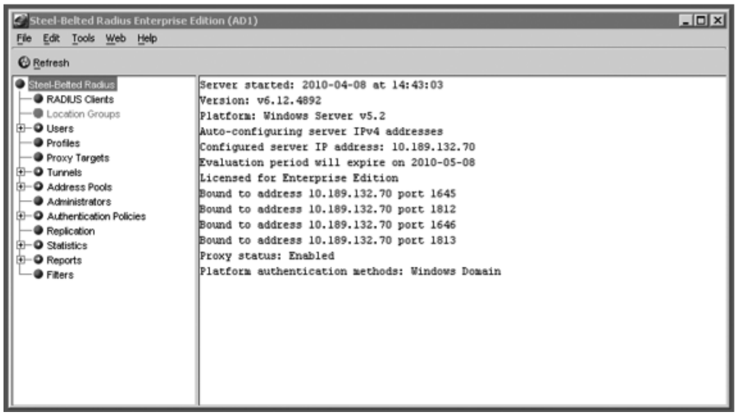
Here are some very straightforward instructions to help if you want to use Steel-Belted RADIUS (SBR) in your network for authentication and authorization tasks, and integrate it with Active Directory. This section complements the configuration snippets from Chapter 5, so feel free to go back and forth as needed to get the best experience.

Installing SBR in an Active Directory server provides seamless integration with the directory schema. After some initial configuration for the NAS client, users, and others, the administrator can continue to manage the user accounts directly from the Windows component "Active Directory Users and Computers."

This may be insufficient for many scenarios and advanced configurations, but the expectation here, as it has been with everything in this Day One book, is that it gets you started, and that you can advance into more complex scenarios from here.

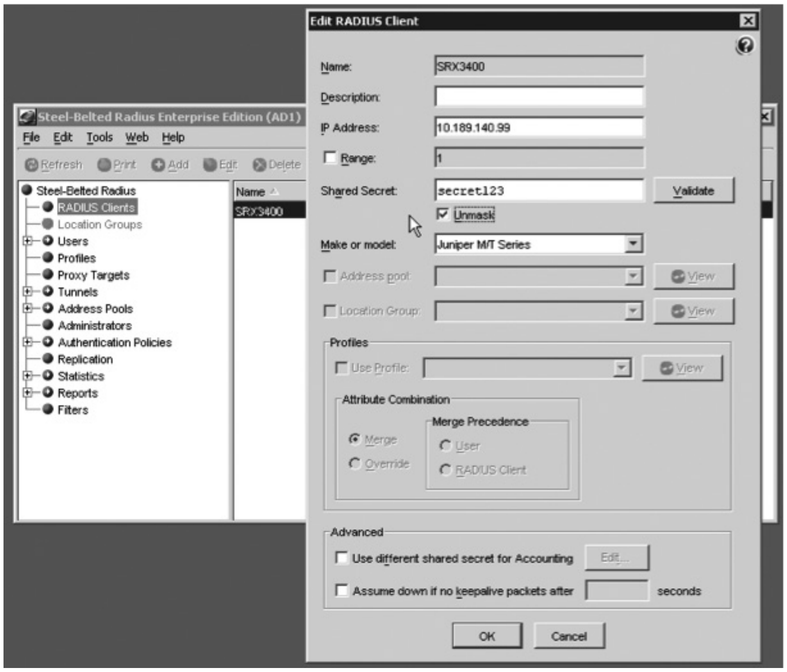
To Configure RADIUS Authentication and SRX Authorization:

1. Connect to the SBR, and lunch the application. The connection is done by pointing a browser to the SBR's IP address on port 1812, like http://[server_IP]:1812. When the application launches it looks something like this:



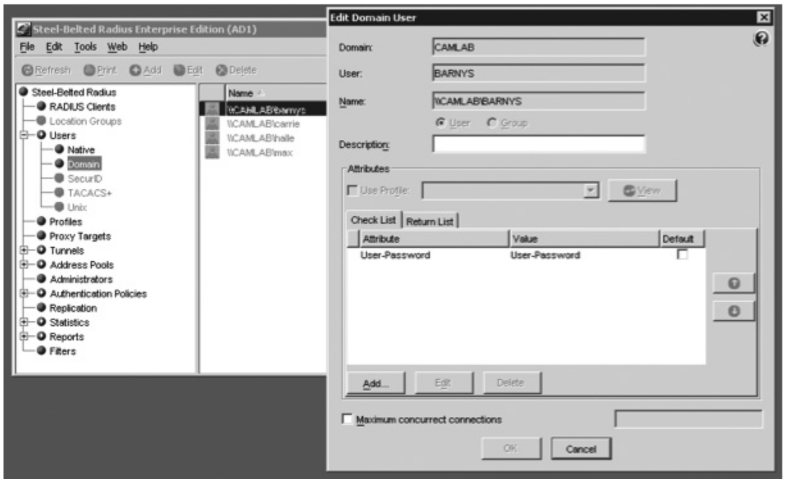
```
Server started: 2010-04-08 at 14:43:03
Version: v6.12.4892
Platform: Windows Server v5.2
Auto-configuring server IPv4 addresses
Configured server IP address: 10.189.132.70
Evaluation period will expire on 2010-05-08
Licensed for Enterprise Edition
Bound to address 10.189.132.70 port 1645
Bound to address 10.189.132.70 port 1812
Bound to address 10.189.132.70 port 1646
Bound to address 10.189.132.70 port 1813
Proxy status: Enabled
Platform authentication methods: Windows Domain
```

2. Add the NAS client, in this case the SRX. Ensure that the share secret matches to the one configured in the firewall. The make and model selected works fine, as the authentication mechanism works the same in M/T Series of routers.



Name: SRX3400
Description:
IP Address: 10.189.140.99
☐ Range: 1
Shared Secret: secret123
☑ Unmask
Make or model: Juniper M/T Series
☐ Address pool:
☐ Location Group:
☐ Use different shared secret for Accounting
☐ Assume down if no keepalive packets after seconds

3. Add users, such as our book's testbed users,: barnys, carrie, halle and max. Notice their names \\CAMLAB\[user] indicates that this can part of the CAMLAB domain. Also, when there is a request from the SRX it asks the SBR server to check for the User-Password against the submitted username.



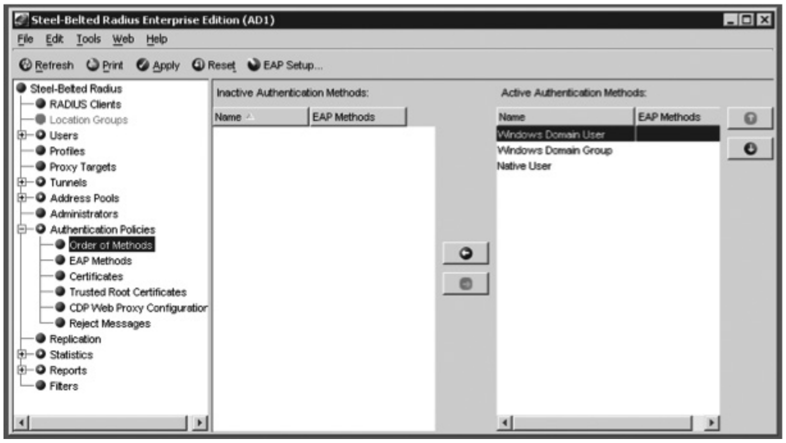
\\CAMLAB\barnys
\\CAMLAB\carrie
\\CAMLAB\halle
\\CAMLAB\max

Domain: CAMLAB
User: BARNYS
Name: \\CAMLAB\BARNYS
[✓ User] Group
Description:

Attribute	Value	Default
User-Password	User-Password	☐

☐ Maximum concurrent connections

4. Change the authentication order so that for incoming requests SBR searches first for the Windows user domain, then for Windows group membership, and lastly for any accounts created in the SBR database.



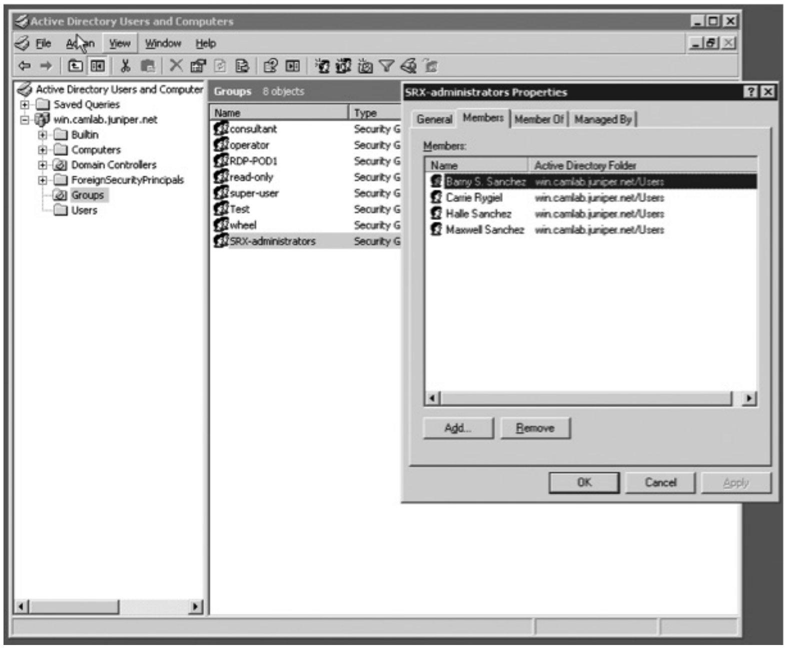
Inactive Authentication Methods:

Name	EAP Methods

Active Authentication Methods:

Name	EAP Methods
Windows Domain User	
Windows Domain Group	
Native User	

5. Ensure that the users in question exist in the Active Directory. To keep things clean, the administrator can choose to create a SRX-administrators group, and make all of the members a part of it.

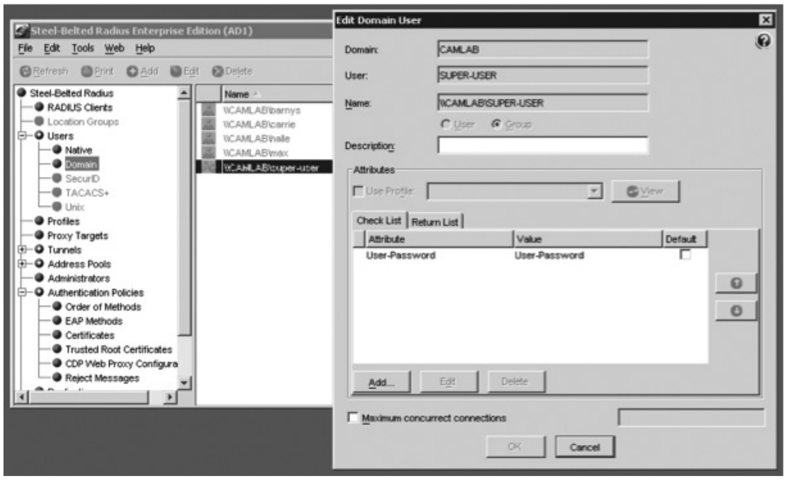

Active Directory Users and Computers
File Action View Window Help
Active Directory Users and Computer
Saved Queries
win.camlab.juniper.net
Builtin
Computers
Domain Controllers
ForeignSecurityPrincipals
Groups
Users
Groups 8 objects
SRX-administrators Properties
General Members Member Of Managed By
Members:
Add... Remove
OK Cancel Apply

Name	Type
consultant	Security G
operator	Security G
RDP-POD1	Security G
read-only	Security G
super-user	Security G
Test	Security G
wheel	Security G
SRX-administrators	Security G

Name	Active Directory Folder
Barry S. Sanchez	win.camlab.juniper.net/Users
Carrie Rygiel	win.camlab.juniper.net/Users
Halle Sanchez	win.camlab.juniper.net/Users
Maxwell Sanchez	win.camlab.juniper.net/Users

To Configure RADIUS Authentication and Authorization:

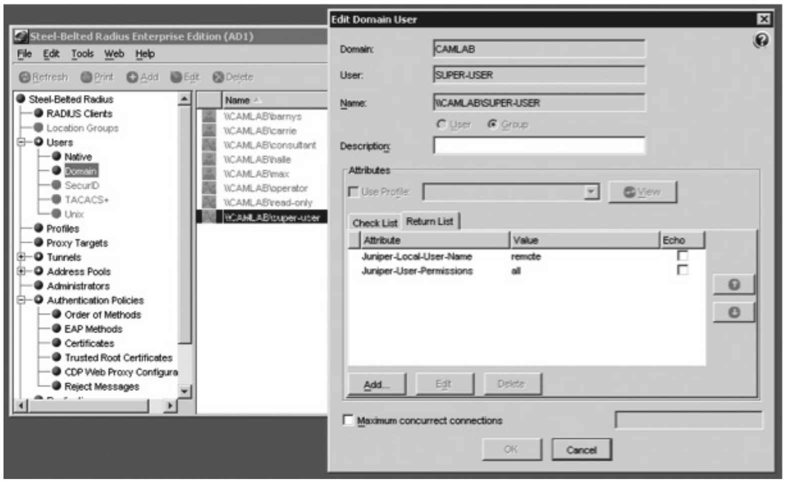
1. To define RADIUS based authentication and authorization, create four user groups in SBR, starting with the group SUPER-USER. Notice that SBR is configured to again check the submitted password against the username supplied.



Domain: CAMLAB
User: SUPER-USER
Name: \\CAMLAB\SUPER-USER
User [✓ Group]
Description:

Attribute	Value	Default
User-Password	User-Password	☐

2. Map the user template *remote* to the SUPER-USER group, and assign all user permissions. In this way a connecting user that belongs to this group in Active Directory, is granted full privileges.

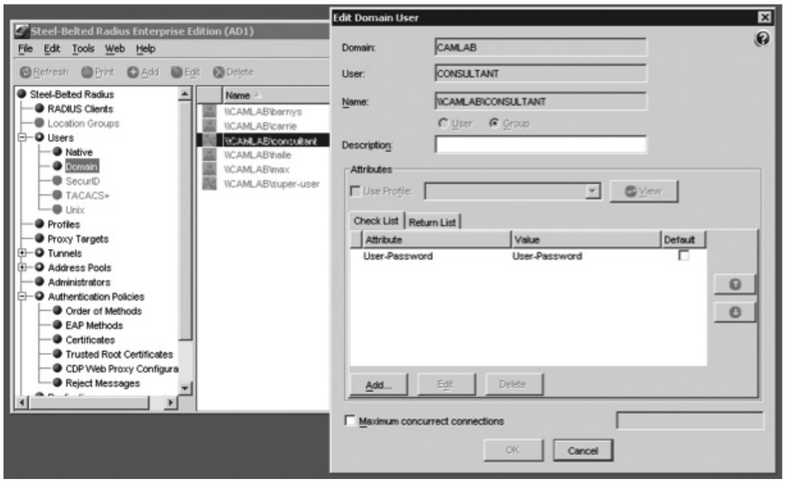


Domain: CAMLAB
User: SUPER-USER
Name: \\CAMLAB\SUPER-USER
User [✓ Group]
Description:

Attribute	Value	Echo
Juniper-Local-User-Name	remote	☐
Juniper-User-Permissions	all	☐

☐ Maximum concurrent connections

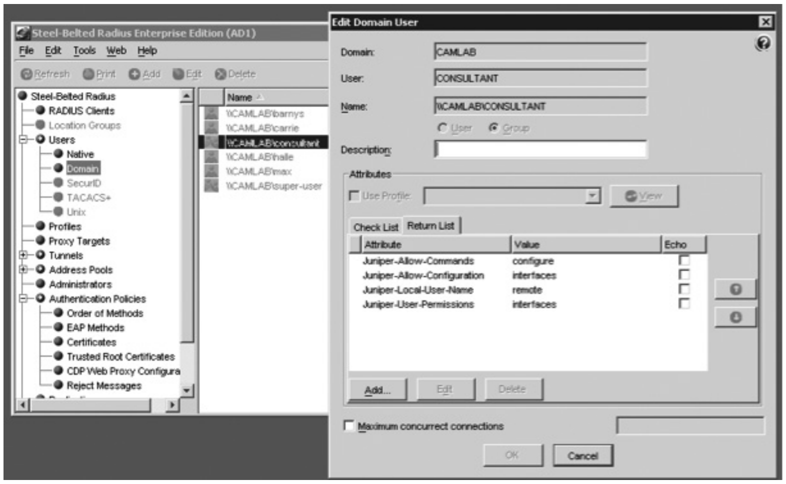
3. The next two screenshots present the same process for the CONSULTANT group. Notice that the return list is more specific, allowing configuration changes only over interfaces.



Domain: CAMLAB
User: CONSULTANT
Name: \\CAMLAB\CONSULTANT
User [✓ Group]
Description:

Attribute	Value	Default
User-Password	User-Password	☐

☐ Maximum concurrent connections



Edit Domain User

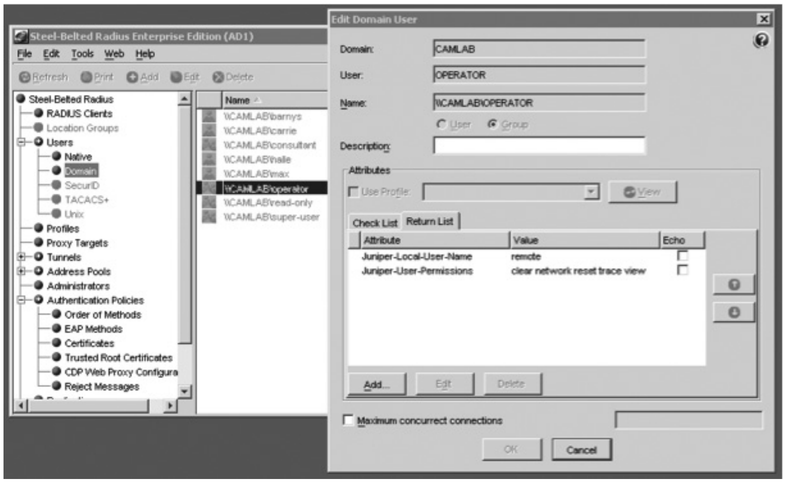
Domain: CAMLAB
User: CONSULTANT
Name: \\CAMLAB\CONSULTANT
User [✓ Group]
Description:

Attributes

Attribute	Value	Echo
Juniper-Allow-Commands	configure	☐
Juniper-Allow-Configuration	interfaces	☐
Juniper-Local-User-Name	remote	☐
Juniper-User-Permissions	interfaces	☐

☐ Maximum concurrent connections

4. The next screenshot is the same process for the READ-ONLY and OPERATOR groups. Again, take note of the permissions for each group.



Edit Domain User

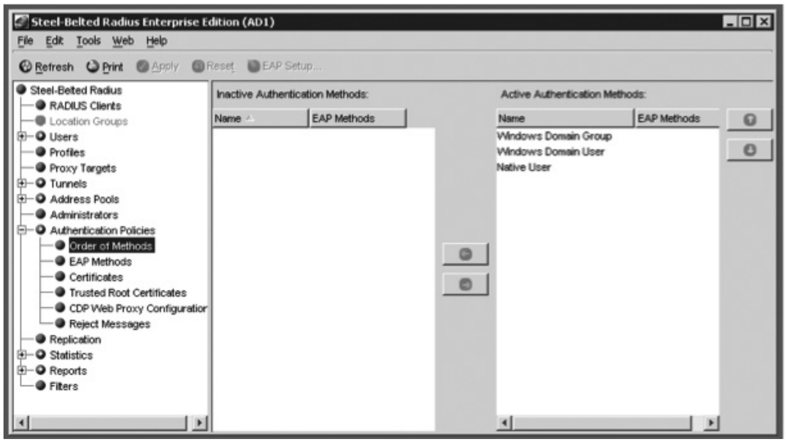
Domain: CAMLAB
User: OPERATOR
Name: \\CAMLAB\OPERATOR
User [✓ Group]
Description:

Attributes

Attribute	Value	Echo
Juniper-Local-User-Name	remote	☐
Juniper-User-Permissions	clear network reset trace view	☐

☐ Maximum concurrent connections

5. Change the authentication order so that now the group membership is searched first, before the individual user accounts.



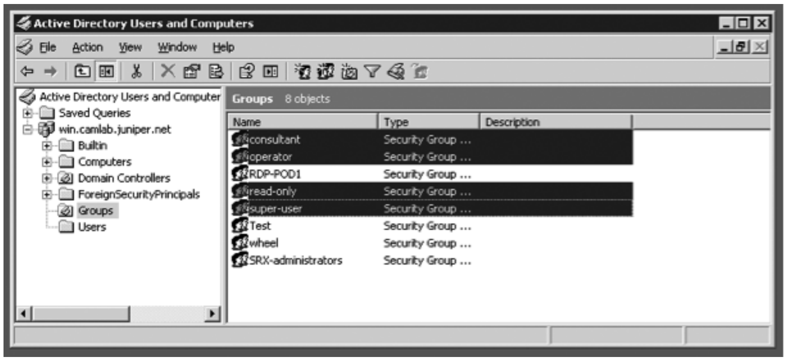
Inactive Authentication Methods:

Name	EAP Methods

Active Authentication Methods:

Name	EAP Methods
Windows Domain Group	
Windows Domain User	
Native User	

6. Make sure that you have the groups created in Active Directory, and that the users belong to their corresponding groups, according to the original intentions.


Active Directory Users and Computers
File Action View Window Help
Active Directory Users and Computer
Saved Queries
win.camlab.juniper.net
Builtin
Computers
Domain Controllers
ForeignSecurityPrincipals
Groups
Users

Groups 8 objects

Name	Type	Description
consultant	Security Group ...	
operator	Security Group ...	
RDP-POD1	Security Group ...	
read-only	Security Group ...	
super-user	Security Group ...	
Test	Security Group ...	
wheel	Security Group ...	
SRX-administrators	Security Group ...	

What to Do Next & Where to Go

www.juniper.net/dayone

If you're reading a print version of this booklet, go here to download the PDF version or find out what other Day One booklets are currently available.

www.juniper.net/junos

Everything you need for Junos adoption and education.

http://forums.juniper.net/jnet

The Juniper-sponsored J-Net Communities forum is dedicated to sharing information, best practices, and questions about Juniper products, technologies, and solutions. Register to participate in this free forum.

www.juniper.net/techpubs

All Juniper-developed product documentation is freely accessible at this site. Find what you need to know about the Junos operating system under each product line.

www.juniper.net/books

Juniper works with multiple book publishers to author and publish technical books on topics essential to network administrators. Check out this ever-expanding list of newly published books including the new SRX-specific *Junos Security*.

www.juniper.net/training/fasttrack

Take courses online, on location, or at one of the partner training centers around the world. The Juniper Network Technical Certification Program (JNTCP) allows you to earn certifications by demonstrating competence in configuration and troubleshooting of Juniper products. If you want the fast track to earning your certifications in enterprise routing, switching, or security use the available online courses, student guides, and lab guides.