

Engineering. Simplicity

DAY ONE: MIGRATING FROM CISCO TO JUNIPER NETWORKS

Book 1: *Migrating From Cisco ASA to Juniper SRX Series*

Book 2: *Migrating From Cisco Catalyst to Juniper EX Series*

Prepare for the future by replacing
your aging Cisco devices with state-
of-the-art Juniper hardware and double
your productivity with this two-book
migration guide for the SRX and EX Series.

By Martin Brown and Rob Jeffery

DAY ONE: MIGRATING FROM CISCO TO JUNIPER NETWORKS

Book 1: Migrating From Cisco ASA to Juniper SRX Series

This *Day One* book walks you step-by-step through a best practice change process that will ease, and actually simplify, a migration from ASA to SRX. It documents a detailed migration plan that will help you familiarize yourself with the Junos OS and the SRX Series. This book also includes dozens of configuration detail comparisons that will make any cutover, in the lab or in production, successful.

"This book is an excellent foundation for migrating from Cisco ASA to Juniper SRX for your organization's next-generation security platform." – Clay Haynes, JNCIE-SEC #69

Book 2: Migrating From Cisco Catalyst to Juniper EX Series

Migrate from Catalyst to Juniper's EX Series in a series of well-defined steps that will prepare you for the future with Juniper's switching platform. Book 2 documents a migration plan that will help you insert the first Junos OS switches into an operational Cisco network.

"Quintessential reading for the engineer migrating from Catalyst to the EX Series. Full of best practices and tips as you upgrade your switching." – Nick Ryce, Senior Network Architect, Fluency, JNCIE-ENT #232

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

Book 1

- Replace aging ASA devices with the SRX Series firewalls
- Compare ASA commands with equivalent Junos OS commands
- Understand the differences between ASA named interfaces and Junos OS zones
- Move from the ASA policy-based VPN towards the SRX Series route-based VPN

Book 2

- Better understand the different interior gateway protocols
- Translate IOS commands to Junos OS commands
- Learn the differences between HSRP and VRRP for gateway redundancy
- Understand how a Virtual Chassis can ease network management

Juniper Books are singularly focused on network productivity and efficiency. Peruse the complete library at www.juniper.net/books.

ISBN 978-1-941441-66-4



9 781941 441664



www.formits.com | tel: 01582 572727

JUNIPER
NETWORKS

AMBASSADOR

Day One: Migrating From Cisco to Juniper Networks

By Martin Brown & Rob Jeffery

<i>Book One: Migrating From Cisco ASA to Juniper SRX Series</i>	5
<i>Chapter 1: Nameifs and Zones</i>	9
<i>Chapter 2: ACEs, ACLs, and Firewall Filters</i>	21
<i>Chapter 3: NAT</i>	31
<i>Chapter 4: Site-to-Site VPN</i>	38
<i>Chapter 5: Device Management</i>	43
<i>Conclusion: The Migration Process</i>	58
 <i>Book Two: Migrating From Cisco Catalyst to Juniper EX Series</i>	59
<i>Chapter 6: IOS to Junos Configuration</i>	64
<i>Chapter 7: VLANs and RVIs</i>	72
<i>Chapter 8: Link Aggregation</i>	80
<i>Chapter 9: Access to Trunk Ports</i>	85
<i>Chapter 10: Redundancy</i>	95
<i>Chapter 11: Security Hardening</i>	101
<i>Conclusion: The Migration Process</i>	105

© 2018 by Juniper Networks, Inc. All rights reserved.
Juniper Networks and Junos are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo and the Junos logo, are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners. Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Published by Juniper Networks Books

Authors: Martin Brown and Rob Jeffery
Technical Reviewers: Clay Haynes, Nick Ryce, Chris Jones
Editor in Chief: Patrick Ames
Copyeditor and Proofer: Nancy Koerbel

ISBN: 978-1-941441-66-4 (paperback)
Printed in the USA by Vervante Corporation.

ISBN: 978-1-941441-67-1 (ebook)

Version History: v1, January 2018
2 3 4 5 6 7 8 9 10

<http://www.juniper.net/dayone>

About the Authors

Martin Brown is a Network Security Engineer for a tier 1 service provider based in the UK, and a Juniper Ambassador with knowledge that covers a broad range of network devices. Martin started his career in IT 20 years ago supporting Macintosh computers, became an MCSE in 1999, and has since progressed to networking, supporting most of the major manufacturers including Cisco, F5, Checkpoint, and of course, Juniper.

Rob Jeffery is a Cyber security Consultant working with resellers, distributors, and vendors, helping them to deliver security solutions and advisory services to large enterprise and public sector organizations. After starting his work life in the hospitality industry, Rob gained a 1st class with hours degree in Network Infrastructure Technologies and Business Computing Solutions, before joining a security VAR and MSSP. Since then he has worked his way from 1st line support up to Technical Director at a security VAR before going solo.

Authors' Acknowledgments

Martin Brown: I would like to thank my friend and colleague Adam Dawson for the awesome save, catching my MacBook Pro just as it fell off the seat, thereby allowing me to continue writing this book, my friend Joy Horton for telling me to relax more and for being a source of inspiration, my fellow Juniper Ambassadors for all their help, advice and support, most of which was related to networking, and to Julie Wider for keeping us in check and for making this all happen.

Rob Jeffrey: Firstly, I would like to thank my co-author Martin for working on this book with me; it is our second book together and I hope in the future we can write more. Secondly I would like to thank Patrick Ames, Editor in Chief at Juniper Networks Books, for constantly pestering me to finish my half. And lastly I would like to thank Julie Wider who runs the Ambassador program at Juniper Networks, without her I would have never had the opportunity to write one book for Juniper let alone two!

Book One: Migrating From Cisco ASA to Juniper SRX Series

Juniper Networks SRX Series Services Gateways

The *SRX Series Services Gateway* is the official name of the SRX Series firewall. This *Day One* book uses the much easier to read generic term, *SRX Series*, to mean any model of the SRX line of service gateway models that you may have in the lab or newly shipped from the factory for installation. Look for other books, manuals, and guides on the specifications of each model of the SRX Series at the Juniper TechLibrary: www.juniper.net/documentation.

What You Need to Know Before Reading This Book

Before reading this book, you should be familiar with the basic administrative functions of the Junos operating system, including the ability to work with operational commands and to read, understand, and change Junos configurations. There are several books in the *Day One* library on learning Junos, at www.juniper.net/dayone.

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

- You have a basic understanding of Internet Protocol version 4 (IPv4).
- You have access to a lab with at least the following components: one Cisco ASA running ASA version 8.3 or higher, and any one of the SRX Series Services Gateways, or even a J-Series router.

It's highly recommended that you peruse the available technical documentation in order to become fully acquainted with the initial configuration process of Junos devices, and to get a better understanding of the configuration process flow. The technical documentation can be located at www.juniper.net/documentation.

What You Will Learn by Reading This Book

- Replace aging ASA devices with the SRX Series firewalls.
- Compare ASA commands with equivalent Junos OS commands.
- Understand the differences between ASA named interfaces and Junos OS zones.
- Move from the ASA policy-based VPN towards the SRX Series route-based VPN.

Day One Junos OS Resources

Several *Day One* books exist for the engineer to better understand the Junos OS, all of them available at <http://www.juniper.net/books>:

- *Exploring the Junos CLI, Second Edition*
- *Routing the Internet Protocol*
- *This Week: Hardening Junos Devices, Second Edition*
- *Finishing Junos Deployments*
- *Junos QoS for IOS Engineers*
- *Junos for IOS Engineers*
- *Deploying Basic QoS*
- *Junos Tips, Techniques, and Templates 2011*

Preface

These days almost every company, organization, and entity has some sort of connection to the global Internet. It is almost impossible to exist without it. And you can make a pretty accurate assumption that any company connected to the Internet will have a firewall in order to protect its network and the data it carries.

But firewalls do get old and it should be pretty obvious to those in IT that while the operating systems and software can be updated, their platforms have a finite life span and will need to be replaced. Such is the case with the many Cisco ASA firewalls. At some point you will replace the ASAs, and when you do we suggest migrating to the Juniper Networks SRX Series Services Gateways. This book will help you to make the transition with a step-by-step, best practice approach, because we, like you, know that such hardware migrations can be a complex and drawn out process.

There are many things you need to consider in such a migration. How easy will it be to replace the hardware? What about the speed and capacity of the hardware? For example, if you are replacing a firewall that allows 100,000 concurrent connections over 1Gb links, it would be unwise to replace it with a smaller 100Mb firewall capable of only 50,000 concurrent connections, regardless of the price.

In the hypothetical test case created for this book, an organization called ACME is replacing their older ASAs, running version 9.2.4 of the ASA software, with a new next-generation firewall, the SRX Series. ACME is preparing for the future, upgrading the perimeter for more cloud-delivered services, and essentially strengthening their security posture.

ACME chose the new SRX Series because its throughput is higher than the ASA, and it is available at a more favorable price. In addition, ACME has several QFX and EX switches in the core and access layers, so the engineers who will be deploying the new firewall already speak Junos, and those who don't can use this book to help them learn how.

Figure P.1 illustrates ACME's network topology. The existing ASA connects to a router, which in turn connects to the Internet. The router is a customer edge (CE) router managed by ACME's service provider, and as such does not utilize a firewall.

The ASA is then connected to the demilitarized zone (DMZ), which contains the web server, an email server, and a proxy server for allowing clients inside the corporate LAN access to the Internet. Devices within the DMZ use RFC 1918 addresses. The ASA is then connected to an inside firewall via a switch. The ASA is then connected to an inside firewall via a switch.

NOTE While ACME has an inside firewall, it is neither an SRX Series nor an ASA, as best practice dictates that when 'dual skinning' an Internet edge, the inside and outside firewalls should be from different vendors, therefore this firewall will not be touched during the migration to the new SRX Series.

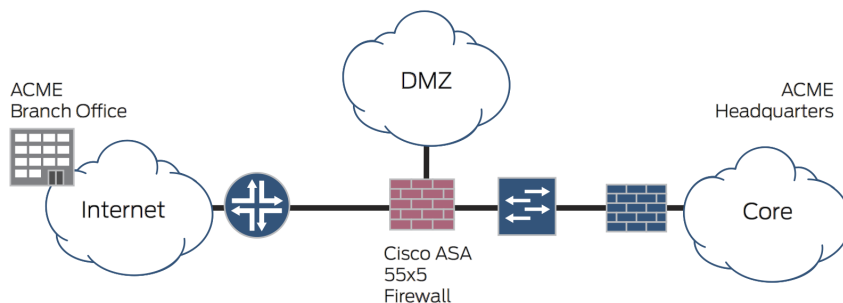


Figure P.1

ACME's Network Topology

In the core of the network are several servers that are used for managing network devices:

- ACS/TACACS
- Syslog
- SNMP
- NTP

Outside the home office network, somewhere on the Internet, ACME has a branch

office shown in Figure P.1. Because this branch is small, the IT director decided a long time ago that the cost of a WAN link was too high, and instead elected to use a site-to-site VPN link spanning the Internet.

Here is the list of what needs to be configured:

- Three interfaces, which must include the same IP addresses that are in place on the ASA.
- Firewall rules, which will need to be converted from ASA to Junos OS.
- Network Address Translators (NATs), which will be converted to allow Internet access to the proxy server, and to allow Internet users to access the corporate web server in addition to sending and receiving emails.
- A site-to-site VPN that needs to be configured with the same policies currently in use.
- In addition, the new SRX Series must be able to connect to the management servers in the data center and must be set to allow only remote access from the management subnet.

NOTE The version of ASA software used in this book is 9.2.4. Note that prior to version 8.3, the NAT statements were very different. The version of Junos OS on the SRX Series is the JTAC version recommended while this book was being written, however, most of the commands used in this book are version neutral, applicable to almost any version of Junos.

There are certain configuration settings that will become more apparent as the book begins to track the migration from the ASA to the shiny new SRX Series. However, for now, relax, kick back, and prepare to learn all about the differences between “Named Interfaces” and “Zones.” Enjoy.

Martin Brown and Rob Jeffery, January 2018

Chapter 1

Nameifs and Zones

Typically, one of the first tasks an engineer performs when configuring network devices, such as a firewall or router, is to assign IP addresses to the relevant interfaces. This may also involve setting speeds, or adding a description, but configuring the interfaces is typically the first task that needs to be completed, because any other changes required on the device – whether they involve creating rules, specifying logging servers, or adding a route – rely on the interfaces being up and configured so an engineer can test whether the configuration is working as expected.

ASAs, however, have another option that needs to be set when configuring interfaces and that is to set the interface name, also known as a *nameif*. When rules, routes, and management servers are added to the configuration, the ASA is told which *named interface* is used to forward the packet, or, in the case of rules, the ASA is told to which named interface the rule is applied.

The SRX Series firewalls do not have a concept of named interfaces. Instead the SRX Series firewalls use *zones* and the firewall rules are applied to the zones. The advantage of zones is that multiple interfaces can be applied to zones if an administrator wishes to do so, although typically only one interface is applied to a zone. The advantage in either case is that it allows an administrator to migrate a zone or a named interface to another interface without having to rewrite a lot of configuration.

The migration begins with the initial ASA configuration to the SRX Series (initial meaning the hostname, domain name, and the interface configuration). Before we begin the initial configuration steps, however, it is important to note that the SRX Series usually includes some default configuration that may not be desirable on your own network, therefore, in our test case, the first task is to remove these default items.

Removing Default Settings

When an SRX Series device is powered on out of the box, or if you use the command `request system zeroize`, the SRX Series adds a few configuration commands by default. These include settings such as creating the default VLAN, adding the interfaces to the default VLAN, setting an IP address on the first interface, enabling a DHCP server, specifying which DNS servers to use, NATs, and the creation of zones and basic policies.

The default SRX Series configuration is as follows. The options in bold are options that should be deleted prior to continuing with the configuration. The option `autoinstallation`, that starts with the fifth line, will be automatically deleted upon the first commit:

```
root> show configuration
## Last commit: 2016-08-26 15:10:48 UTC by root
version 12.1X46-D40.2;
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;
        xnm-clear-text;
        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;
        }
    }
}
```

11 Removing Default Settings

```

}
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 {
    ge-0/0/0 {
        unit 0;
    }
    ge-0/0/1 {
        unit 0 {
            family ethernet-switching {
                vlan {
                    members vlan-trust;
                }
            }
        }
    }
}

```

<snip> #All remaining interfaces are part of the same VLAN as interface ge-0/0/1

```

    vlan {
        unit 0 {
            family inet {
                address 192.168.1.1/24;
            }
        }
    }
}
protocols {
    stp;
}
security {
    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;
}
}
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;
                    }
                }
            }
        }
    }
}
policies {
    from-zone trust to-zone untrust {
        policy trust-to-untrust {
            match {
                source-address any;
                destination-address any;
                application any;
            }
            then {
                permit;
            }
        }
    }
}
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 {

```

```

    }
}
vpls {
    vlan-trust {
        vlan-id 3;
        l3-interface vpls.0;
    }
}
system-services {
    dhcp;
    tftp;
}
}
```

Zones and policies are discussed in more detail in Chapter 2, and the VLAN `vlan-trust` will be deleted later in this chapter, but for now, let's delete the Dynamic Host Configuration Protocol (DHCP) server configuration and remove the DNS servers. This is done by entering the following commands:

```
delete system name-server
delete system services dhcp
```

TIP If, after entering these commands, an error appears stating that the command isn't recognized, chances are the user is in the default "shell" mode. This is typical when a user logs in as the default "root" user. To enter exec mode, use the `cli` command. To enter configuration mode, use the `configure` command.

Once the DNS and DHCP information has been loaded use the following command:

```
delete security nat
```

NOTE New NAT statements that are more relevant to our situation will be discussed and configured in Chapter 3.

Setting the Hostname

After clearing these settings, the hostname can be set, along with the domain name. The ASA sets these options by using the following commands:

```
hostname ACME-INTERNET-FW
domain-name ACME.com
```

To set the same option on the SRX Series, use the following:

```
set system host-name ACME-INTERNET-FW
set system domain-name ACME.com
```

It may seem trivial to set the host name, but the SRX Series uses the hostname in the prompt, so engineers can know quite easily which SRX Series device they are connected to, and which one they are adding a configuration to. This might seem

redundant if you're in the lab and the SRX Series is right next to your laptop, but it's very useful if you are configuring the SRX Series via a terminal server. Including the hostname in the prompt provides a simple sanity check that can prevent you from configuring the wrong device and a lot of embarrassment.

Specifying IP Addresses

Now that the basic configuration steps have been performed and the unwanted default configuration has been removed, let's turn our attention to configuring the IP address. Figure 1.1 depicts the Layer 3 topology with the subnets currently connected to the ASA, and which will be connected soon to the SRX Series.

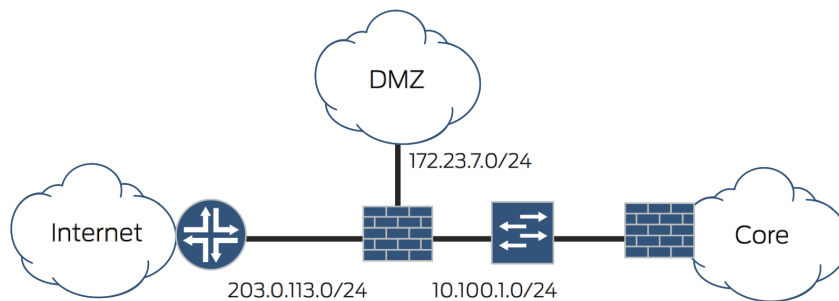


Figure 1.1 Layer 3 Topology

You can see that the ASA currently has three interfaces configured, and the interface configuration on the ASA reveals the following information:

```
interface Gi0/0
nameif INTERNET
security-level 0
ip address 203.0.113.10 255.255.255.0

interface Gi0/1
nameif DMZ
security-level 50
ip address 172.23.7.1 255.255.255.0

interface Gi0/2
nameif INSIDE
security-level 100
ip address 10.100.1.1 255.255.255.0

mtu INTERNET 1500
mtu INSIDE 1500
mtu DMZ 1500
```

The SRX Series to be configured in this scenario will use the same interfaces: ge-0/0/0, ge-0/0/1, and ge-0/0/2. The addresses assigned to these interfaces will also remain the same. The difference, however, is that the ASA uses named interfaces or nameifs. The SRX Series doesn't have any concept of named interfaces and instead uses zones.

The ASA also has the option `mtu <interface-name> 1500` configured, therefore care should be taken to include this information on the SRX Series.

One major difference between ASA and SRX Series interfaces is that almost all Junos devices require the IP addresses or Ethernet switching information to be added to the sub-interfaces, regardless of whether or not only one sub-interface is in use. The default sub-interface is called *unit 0* and it is reflected in the following configuration, which sets the maximum transmission unit (MTU) size followed by the IP address:

```
set interfaces ge-0/0/0 unit 0 family inet mtu 1500
set interfaces ge-0/0/0 unit 0 family inet address 203.0.113.10/24
set interfaces ge-0/0/1 unit 0 family inet mtu 1500
set interfaces ge-0/0/1 unit 0 family inet address 172.23.7.1/24
set interfaces ge-0/0/2 unit 0 family inet mtu 1500
set interfaces ge-0/0/2 unit 0 family inet address 10.100.1.1/24
```

SRX Series interfaces separate Layer 2 and Layer 3 information under the option `family`. For Layer 3 IPv4 information, this is placed under `inet`. IPv6 information is placed under `inet6` and Ethernet switching is placed under `ethernet-switching`. On some SRX Series, some of the interfaces are added to the default VLAN. As the interfaces in this SRX Series have been assigned an IP address under `family inet`, the `ethernet-switching` option needs to be removed by entering the following:

```
delete interfaces ge-0/0/1.0 family ethernet-switching
delete interfaces ge-0/0/2.0 family ethernet-switching
```

If these options were not removed, Junos would display an error similar to the one cited here, stating that if the `family ethernet-switching` option is configured on an interface, then no other family can be configured at the same time:

```
root@ACME-INTERNET-FW# commit
[edit interfaces ge-0/0/2 unit 0]
'family'
```

When the `ethernet-switching` family is configured on an interface, no other family type can be configured on the same interface:

```
error: configuration check-out failed
```

Configuring Zones in Junos OS

In the default configuration, Junos creates two zones on the SRX Series. These zones are “trust” and “untrust.” Initially these zones are really just labels, and even after interfaces are assigned to the zones they technically do nothing until a policy or NAT is applied to the zone pairs. The behavior is similar to the ASA having named interfaces.

NOTE Until a rule is applied to the named interface, the ASA has the option to use security levels, with traffic coming from a higher level interface being able to access devices on a lower level interface, but dropping traffic going from a lower level interface to a higher level interface. The SRX Series does not use security levels and as a result, rules always need to be applied.

While you could just delete the zones and then create new zones, it actually requires fewer steps to rename the existing zones and then create our third zone thereafter. To rename a zone, an administrator simply needs to use the `rename` command; in this instance, rename the zone *trust* to the zone name *INSIDE*:

```
rename security zones security-zone trust to security-zone INSIDE
```

Interfaces can then be applied to the zone like so:

```
set security zones security-zone INSIDE interfaces ge-0/0/2.0
```

And the default VLAN can be removed from the zone like this:

```
delete security zones security-zone INSIDE interfaces vlan.0
```

TIP Note that *INSIDE* is in all upper case letters. This is a common best practice within the industry because uppercase stands out more when reading a device configuration, and any engineer looking at the configuration will know that uppercase is a label, whereas any lowercase will be a command or an option.

You may also notice that this is in contrast to the zone names that are applied by default to the SRX Series, as these are entered in lower case. This is another reason why the default labels need to be changed, so that they become uppercase.

Next, rename the zone *untrust* to *INTERNET* using the commands similar to those you used when the zone “*trust*” was renamed:

```
rename security zones security-zone untrust to security-zone INTERNET
```

By default, Junos allows Trivial File Transfer Protocol (TFTP) and DHCP traffic into the device from the *ge-0/0/0.0* interface. As this is the Internet-facing interface, this option needs to be removed:

```
delete security zones security-zone INTERNET interfaces ge-0/0/0.0 host-inbound-traffic
```

To create the zone “DMZ” and assign interface *ge-0/0/1.0* to the zone, run the following:

```
set security zones security-zone DMZ interfaces ge-0/0/1.0
```

DoS Protection

When an SRX Series is booted to factory defaults, the Junos OS creates a rather useful piece of configuration called a *screen*. This screen will attempt to protect the SRX Series from DoS attacks such as the “Ping of Death” and “Syn-Flood” attacks. The default screen settings look like this:

```
root@ACME-INTERNET-FW> show configuration security 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;
    }
}
```

While it’s a very good idea to keep this option, it may also be a good idea to rename this screen so that it matches the zone name. There is no harm in keeping the screen name as default and the change is purely cosmetic. Should you want to change the name, then this can be done with the same `rename` command used previously:

```
rename security screen ids-option untrust-screen to ids-option INTERNET-SCREEN
```

The screen should be applied to the zone “INTERNET” to take into account the new name:

```
set security zones security-zone INTERNET screen INTERNET-SCREEN
```

Finally, in order to stop an error which prevents a successful commit from occurring, the existing policy that allowed traffic from the zone “trust” to the zone “un-trust”, needs to be renamed to take into account the new zone names:

```
rename security policies from-zone trust to-zone untrust to from-zone INSIDE to-zone INTERNET
```

The policy itself will be removed, and a more relevant policy will be added in Chapter 2.

Removing the Default VLAN

If your SRX Series has created a default VLAN, then now would be a good time to remove the Layer 3 interface that is associated with it. Later on, if your organization's security policy dictates, you can create an unused VLAN, assign all unused ports to this unused VLAN, and completely remove the default VLAN. For now, however, only the Layer 3 interface will be removed:

```
delete interfaces vlan.0
```

Even though the Layer 3 interface has been removed, the default VLAN will still reference this interface under the VLAN configuration, therefore it should be removed as well, otherwise the commit will fail:

```
delete vlans vlan-trust l3-interface
```

Creating a Default Route

Once the interfaces have been configured, the next step should be to add routing information to the device. In this case, the device is not using a dynamic routing protocol, and as such, all routes are static. The ASA has a default route facing out the INTERNET interface, another route summarizing the internal corporate networks into a single 10.100.0.0/16 route, and a third route to the subnet 10.255.1.0/24:

```
route INTERNET 0.0.0.0 0.0.0.0 203.0.113.1 1
route INSIDE 10.100.0.0 255.255.0.0 10.100.1.2
route INSIDE 10.255.1.0 255.255.255.0 10.100.1.2
```

Junos sets these options under the `routing-options` hierarchy. With Junos there is no need to specify the exit interface, just the next hop, and Junos will figure out for itself which interface that is:

```
set routing-options static route 0.0.0.0/0 next-hop 203.0.113.1
set routing-options static route 10.100.0.0/16 next-hop 10.100.1.2
set routing-options static route 10.255.1.0/24 next-hop 10.100.1.2
```

Setting the Root Password

The SRX Series, like the ASA, has no default password set on the device. Unlike the ASA, the SRX Series does create a user default called `root` and the password is so important that the SRX Series will not let you save the configuration until you have set the password for the root user. The command to set the password for the root user is:

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

After you've set the password for the root user, Junos will prompt you to enter the password and then ask you to confirm the password:

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

It is of the utmost importance that the password used is a complex password, which only a few administrators know, because root user is allowed full access to the entire system, including the underlying operating system.

TIP It's best practice to create a second user immediately, and to use that second user to administer your device. Never use the root user unless it is absolutely necessary.

Saving Changes to the SRX Series

After all the changes have been applied, they need to be saved to the device by using the command `commit`. There are several options available to you when applying this command. The first option is to `commit check`, which asks Junos to do a sanity check on the configuration without applying the configuration itself.

The next option is `commit confirmed`. This tells Junos to roll back the configuration to the previous state in a time that is specified. This is very useful, for example, when an engineer accidentally locks himself out by adding a rule that prevents management traffic.

If you have finished the configuration process, then `commit and-quit` will tell Junos to exit the configuration mode. Now let's just use the single `commit`:

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

```
[edit]
root@ACME-INTERNET-FW#
```

Note how the prompt has changed from `root#` to `root@ACME-INTERNET-FW#`, displaying the host name of the device.

Confirming the Changes Have Been Successful

After making changes such as the ones we've just addressed, it is best practice to check your work, ensuring the changes you have made have had the desired effect on your new SRX Series. You'll need to come back and perform a diagnosis anyway, so it's a good idea to check your work now before continuing.

Let's check whether the interfaces are up and configured with the correct

addresses. The `show interfaces terse` command can help with this. Here, a `match` and `exclude` statement has been used to remove the loopback interfaces and `sp` interfaces, and display only the relevant `ge` interfaces:

```
root@ACME-INTERNET-FW> show interfaces terse | match inet | exclude lo0 | exclude s
ge-0/0/0.0      up    up    inet    203.0.113.10/24
ge-0/0/1.0      up    up    inet    172.23.7.1/24
ge-0/0/2.0      up    up    inet    10.100.1.1/24
```

Next, the routing table should be checked for the presence of the static route, ensuring it points to the correct next hop:

```
root@ACME-INTERNET-FW> show route protocol static

inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0      *[Static/5] 01:19:11
                > to 203.0.113.1 via ge-0/0/0.0
```

Finally, let's perform a simple ping test to ensure the SRX Series can reach the next hop. The issue is, the current firewall is live and you cannot have the SRX Series on the same subnet as the ASA or you will suffer from an IP address conflict. You could, of course, use another address, but if your ISP will not issue another public IP address for test purposes, then this will not be an option. So, in this case, the SRX Series has been connected to a lab environment in order to perform limited testing prior to migrating onto the live network:

```
root@ACME-INTERNET-FW> ping 203.0.113.1
PING 203.0.113.1 (203.0.113.1): 56 data bytes
64 bytes from 203.0.113.1: icmp_seq=0 ttl=255 time=3.426 ms
64 bytes from 203.0.113.1: icmp_seq=1 ttl=255 time=3.221 ms
64 bytes from 203.0.113.1: icmp_seq=2 ttl=255 time=3.347 ms
^C
--- 203.0.113.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 3.221/3.331/3.426/0.084 ms
```

As you can see, the ping test has been successful, so let's move on to the next phase of the migration: creating the firewall rules.

Chapter 2

ACEs, ACLs, and Firewall Filters

Now that the interfaces have been given IP addresses, the SRX Series is pretty much behaving like a router you might have at home. All traffic is being allowed out but nothing is being allowed back in. To correct this, policies need to be added that mirror those currently configured on the ASA.

But before examining the existing policies, let's first look at what the policies are trying to achieve.

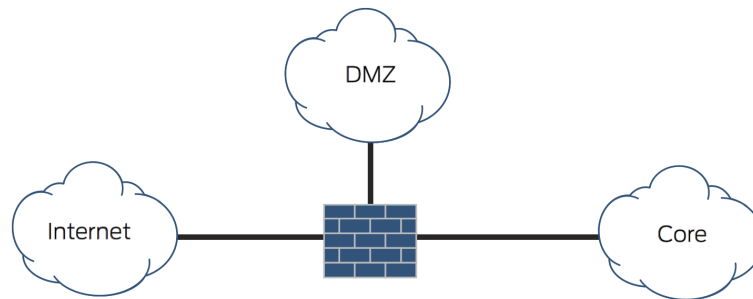


Figure 2.1 ACME Firewall Areas

ACME has divided their Internet-facing firewall into three areas. The first area is the interface connected to the Internet. As established in Chapter 1, this interface has a name of *INTERNET*. The second interface has the name of *DMZ* and finally the inside interface has the name of *INSIDE*. Figure 2.1 shows a representation of this configuration.

In order to restrict traffic flows, Access Control Lists (ACLs) have been applied to all three interfaces. ACLs are made up of multiple entries called Access Control Entries (ACEs). In order to make administration easier, the ACLs have been given the same name as the interface to which they are attached, for example the ACL applied to the INTERNET interface has the name INTERNET_ACL. Figure 2.2 shows the five VLANs that are reachable via the Inside interface.

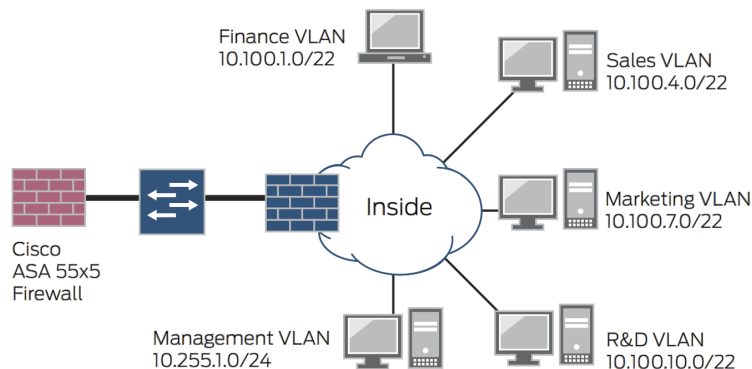


Figure 2.2 ACME Inside VLANs

Included are four VLANs for various departments within the company, and the Management VLAN for managing network devices and servers.

The second interface is connected to the DMZ. In the DMZ are three hosts – a web server, an email server, and a proxy server – as shown in Figure 2.3.

All hosts on the INSIDE interface need to be able to access all of these servers, but only on the applicable ports; for example, the web server would only expect HTTP, HTTPS, and FTP traffic.

At ACME, the hosts on the inside are only allowed to access the Internet via the proxy server, with the exception of the Management VLAN. It seems the network engineers have convinced their manager to allow unrestricted access to the Internet directly from this VLAN without going through the proxy. The reason for this is that if the proxy server went down they could still download patches and use the Internet to search for fixes. (This happens a lot.) Anyway, Figure 2.4 depicts the VLAN setup.

You can see in Figure 2.4 that the interface connected to the Internet must allow for traffic coming from the branch office. This traffic should be coming via a VPN tunnel, but nonetheless, it has to be allowed in. Figure 2.4 shows the branch and the subnet in use. The policy must also allow Internet users to access the corporate web server and email servers to connect to the corporate email server.

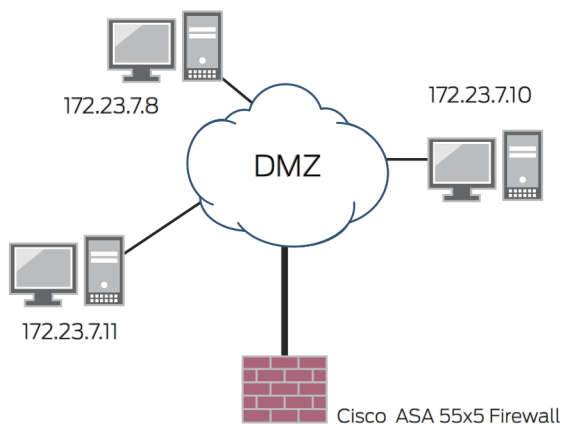


Figure 2.3 ACME DMZ Hosts

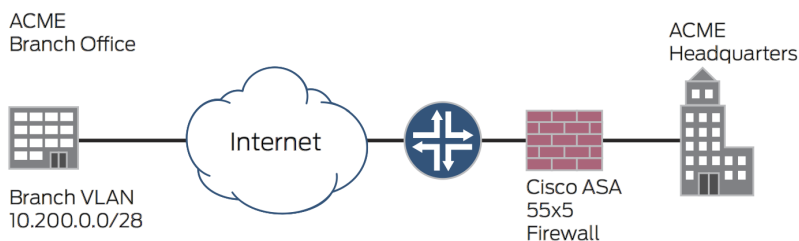


Figure 2.4 ACME Branch and VLAN

ASA Objects and Object Groups

In order to allow ACEs to reference a name, *objects* and *object groups* are created on the ASA. The advantage of this is that the objects and object groups can be updated without rewriting an ACL that is working perfectly fine. The first object group on the ASA contains the VLANs in the corporate network:

```
object-group network CORP-VLANS
network-object 10.100.1.0 255.255.252.0
network-object 10.100.4.0 255.255.252.0
network-object 10.100.7.0 255.255.252.0
network-object 10.100.10.0 255.255.252.0
```

Next are the objects that refer to the servers in the DMZ, the email server, the web server, and the proxy server:

```
object network E-MAIL_SERVER
host 172.23.7.8
```

```
object network WEB-SERVER  
host 172.23.7.11
```

```
object network PROXY-SERVER  
host 172.23.7.10
```

Next is the VLAN for the management network:

```
object network MANAGEMENT-NETWORK  
subnet 10.255.1.0 255.255.255.0
```

And the last object refers to the VLAN in use in the branch office:

```
object network MK-BRANCH  
subnet 10.200.0.0 255.255.240.0
```

Creating Objects in Junos

Junos OS doesn't use objects – it uses something similar called address books, which can be configured globally or can be configured under each zone. The following example would create an address book entry for a database server called DB-SERVER, with an address of 192.168.1.3 in the global address book:

```
set security address-book global address DB-SERVER 192.168.1.3/32
```

To create the same entry under the zone SERVERS, the command would be as follows:

```
set security zones security-zone SERVERS address-book address DB-SERVER 192.168.1.3/32
```

TIP Note that the prefix of /32 is at the end of the command. This denotes that the entry is a host entry. If the entry was for a whole subnet, then the keyword `wildcard-address` would be used along with a different prefix such as /24. Alternatively, an address range from a start IP address to an end IP address can also be specified.

The question you are undoubtedly asking is: *what is the advantage of using a global address book entry as opposed to a zone-based entry?* The answer is quite simple. If an entry is added to the global address book, then this entry is available to any policy created later on. Zone-based entries are only available under that policy. The other reason to use global entries is because address book entries under zones cannot be used in NAT commands, whereas global entries are allowed.

NAT statements are used in Chapter 3, so it would be better to specify global address book entries. Creating the address book entries for the servers, the management VLAN, and the branch VLAN is relatively straightforward. To create these address book entries to match the ASA objects and groups, use the following commands:

```
set security address-book global address PROXY-SERVER 172.23.7.10/32
set security address-book global address WEB-SERVER 172.23.7.11/32
set security address-book global address E-MAIL-SERVER 172.23.7.8/32

set security address-book global address MANAGEMENT-NETWORK wildcard-address 10.255.1.0/24
set security address-book global address MK-BRANCH wildcard-address 10.200.0.0/28
```

Creating the address book entry for the corporate VLANs is slightly different, as on the ASA the object group could be created and the individual subnets could be added as required. Unfortunately, address book entries only allow a single host or subnet. So thankfully there is a simple solution – create a separate VLAN address book, like this:

```
set security address-book global address FINANCE-VLAN wildcard-address 10.100.1.0/22
set security address-book global address SALES-VLAN wildcard-address 10.100.4.0/22
set security address-book global address MARKETING-VLAN wildcard-address 10.100.7.0/22
set security address-book global address RandD-VLAN wildcard-address 10.100.10.0/22
```

Then assign these address book entries into an address-set called CORP-VLANS using the following commands:

```
set security address-book global address-set CORP-VLAN address SALES-VLAN
set security address-book global address-set CORP-VLAN address RandD-VLAN
set security address-book global address-set CORP-VLAN address MARKETING-VLAN
set security address-book global address-set CORP-VLAN address FINANCE-VLAN
```

When it comes to writing policies, by using an address-set you still need only to reference CORP-VLANS, as opposed to individual VLANs. You can then add to this address-set, or update individual address book entries, as needed.

ASA ACLs and ACEs

As mentioned at the beginning of this chapter, the ASA currently has three ACLs configured to use with the firewall policies. These have the names of *INSIDE_acl*, *DMZ_acl*, and *INTERNET_acl*. The ACEs that make up the ACL *INSIDE_acl* allow access from the inside VLANs to the proxy, Web, and email servers. The management VLAN has unrestricted access to any IPv4 subnet:

```
access-list INSIDE_acl extended permit tcp object-group CORP-VLANS object E-MAIL_SERVER eq smtp
access-list INSIDE_acl extended permit tcp object-group CORP-VLANS object PROXY-SERVER eq http
access-list INSIDE_acl extended permit tcp object-group CORP-VLANS object PROXY-SERVER eq https
access-list INSIDE_acl extended permit tcp object-group CORP-VLANS object PROXY-SERVER eq ftp
access-list INSIDE_acl extended permit tcp object-group CORP-VLANS object WEB-SERVER eq http
access-list INSIDE_acl extended permit tcp object-group CORP-VLANS object WEB-SERVER eq https
access-list INSIDE_acl extended permit ip object-group MANAGEMENT-NETWORK any4
access-list INSIDE_acl extended deny ip any any log
```

ACLs have an implicit deny at the end – when there are no entry matches the traffic is automatically denied. This traffic is not automatically logged, however, so at the end of the ACLs you may notice `deny ip any any log`, which tells the ASA to log any access attempts that were denied.

The access list associated with the DMZ interface needs to allow the proxy server access to the Internet because it is furnishing Internet access requests on behalf of the clients on the INSIDE. The three servers in the DMZ also need to access servers in the Management VLAN for the purposes of time synchronization, sending syslog messages, and sending SNMP traps, therefore the ACL DMZ_acl is configured as follows:

```
access-list DMZ_acl extended permit tcp object E-MAIL_SERVER any4 eq smtp
access-list DMZ_acl extended permit tcp object PROXY-SERVER any4 eq http
access-list DMZ_acl extended permit tcp object PROXY-SERVER any4 eq https
access-list DMZ_acl extended permit tcp object PROXY-SERVER any4 eq ftp
access-list DMZ_acl extended permit tcp object PROXY-SERVER object MANAGEMENT-NETWORK eq snmp
access-list DMZ_acl extended permit tcp object E-MAIL_SERVER object MANAGEMENT-NETWORK eq snmp
access-list DMZ_acl extended permit tcp object WEB-SERVER object MANAGEMENT-NETWORK eq snmp
access-list DMZ_acl extended permit udp object PROXY-SERVER object MANAGEMENT-NETWORK eq ntp
access-list DMZ_acl extended permit udp object E-MAIL_SERVER object MANAGEMENT-NETWORK eq ntp
access-list DMZ_acl extended permit udp object WEB-SERVER object MANAGEMENT-NETWORK eq ntp
access-list DMZ_acl extended permit udp object PROXY-SERVER object MANAGEMENT-NETWORK eq syslog
access-list DMZ_acl extended permit udp object E-MAIL_SERVER object MANAGEMENT-NETWORK eq syslog
access-list DMZ_acl extended permit udp object WEB-SERVER object MANAGEMENT-NETWORK eq syslog
```

Finally, the ACL INTERNET_acl needs to allow access from Internet users to the corporate web server, access from email servers to the corporate email server, and to allow traffic from the branch office to enter. Therefore, it is configured like this:

```
access-list INTERNET_acl extended permit tcp any4 object WEB-SERVER eq http
access-list INTERNET_acl extended permit tcp any4 object WEB-SERVER eq https
access-list INTERNET_acl extended permit tcp any4 object WEB-SERVER eq ftp
access-list INTERNET_acl extended permit tcp any4 object E-MAIL_SERVER eq smtp
access-list INTERNET_acl extended permit icmp object MK-VPN object-group CORP-VLANS
access-list INTERNET_acl extended permit ip object MK-VPN object-group CORP-VLANS
access-list INTERNET_acl extended permit tcp object MK-VPN object PROXY-SERVER eq http
access-list INTERNET_acl extended permit tcp object MK-VPN object PROXY-SERVER eq https
access-list INTERNET_acl extended permit tcp object MK-VPN object PROXY-SERVER eq ftp
access-list INTERNET_acl extended deny ip any any log
```

The ASA is told to apply ACLs to the named interfaces using the following commands:

```
access-group INTERNET_acl in interface INTERNET
access-group DMZ_acl in interface DMZ
access-group INSIDE_acl in interface INSIDE
```

Junos OS Firewall Policies

Instead of using ACLs in the Junos OS, the SRX Series uses security policies. During the creation of a policy, the from-zone and to-zone are specified, meaning that there is no need to bind the policy to the interface, Junos does this already. What you do need to do is create the policy.

Policies are divided into terms. Terms are processed in sequence from the top to the bottom, rather like an ACL. When Junos successfully matches traffic with a

term, Junos will then perform whatever action has been set for that term.

The process for configuring Junos security policies is:

- Specify the from-zone and to-zone to tell the SRX Series which zone the traffic is coming from and which zone it is going to.
- Set a name for the first policy term, such as `Allow-Internet`.
- Use a match statement to tell Junos what to match against – the source, destination, and application are all required information. The keyword `any` is perfectly acceptable.
- End the policy with a then statement to tell Junos what action to take – it can be a permit, deny, reject, log, or count.
- Once the commit is applied, the policy automatically takes effect.

An example of a configured policy looks similar to the following example:

```
[edit security policies]
admin@CORP-SRX# show
from-zone TRUSTED to-zone INTERNET {
  policy ALLOW-INTERNET {
    match {
      source-address CORP-NETWORK;
      destination-address any;
      application any;
    }
    then {
      permit;
    }
  }
}
```

When the interfaces were configured in Chapter 1, the existing policy was renamed in order to avoid an error upon commit. Therefore, the first thing that needs to be done is to remove the default policy:

```
delete security policies from-zone INSIDE to-zone INTERNET policy trust-to-untrust
```

There are now no policies configured in the SRX Series, and all in all, six separate policies need to be created. Let's begin by creating the first policy for traffic going from the zone `INSIDE` to the zone `DMZ`.

TIP Bear in mind that the ACLs on the ASA would be for all traffic, whereas with the SRX Series, only traffic going between those zones need be specified.

The first policy should allow HTTP, HTTPS, and FTP traffic going from the corporate VLANs to the proxy and web servers, then allow SMTP traffic from the corporate VLANs to the email server, and finally allow the management VLAN to be able to access any address:

```

set security policies from-zone INSIDE to-zone DMZ policy WEB-AND-PROXY match source-address CORP-
VLAN
set security policies from-zone INSIDE to-zone DMZ policy WEB-AND-PROXY match destination-
address PROXY-SERVER
set security policies from-zone INSIDE to-zone DMZ policy WEB-AND-PROXY match destination-
address WEB-SERVER
set security policies from-zone INSIDE to-zone DMZ policy WEB-AND-PROXY match application junos-http
set security policies from-zone INSIDE to-zone DMZ policy WEB-AND-PROXY match application junos-
https
set security policies from-zone INSIDE to-zone DMZ policy WEB-AND-PROXY match application junos-ftp
set security policies from-zone INSIDE to-zone DMZ policy WEB-AND-PROXY then permit

set security policies from-zone INSIDE to-zone DMZ policy E-MAIL match source-address CORP-VLAN
set security policies from-zone INSIDE to-zone DMZ policy E-MAIL match destination-address E-MAIL-
SERVER
set security policies from-zone INSIDE to-zone DMZ policy E-MAIL match application junos-smtp
set security policies from-zone INSIDE to-zone DMZ policy E-MAIL then permit

set security policies from-zone INSIDE to-zone DMZ policy MANAGEMENT match source-
address MANAGEMENT-NETWORK
set security policies from-zone INSIDE to-zone DMZ policy MANAGEMENT match destination-address any-
ipv4
set security policies from-zone INSIDE to-zone DMZ policy MANAGEMENT match application any
set security policies from-zone INSIDE to-zone DMZ policy MANAGEMENT then permit

```

Notice that the web server and proxy server are grouped together under the same policy term. It is perfectly acceptable to group multiple destinations together under one policy term – Junos treats these as OR as opposed to AND. The application sets the port, and again, these can be grouped together under the same policy term and would be treated as an OR statement..

The next policy should allow traffic from the Management VLAN to any address on the Internet. At the end, the administrator needs to specify a *deny* and to log when the session is initialized:

```

set security policies from-zone INSIDE to-zone INTERNET policy ALLOW-MANAGEMENT match source-
address MANAGEMENT-NETWORK
set security policies from-zone INSIDE to-zone INTERNET policy ALLOW-MANAGEMENT match destination-
address any-ipv4
set security policies from-zone INSIDE to-zone INTERNET policy ALLOW-
MANAGEMENT match application any
set security policies from-zone INSIDE to-zone INTERNET policy ALLOW-MANAGEMENT then permit

set security policies from-zone INSIDE to-zone INTERNET policy DENY match source-address any-ipv4
set security policies from-zone INSIDE to-zone INTERNET policy DENY match destination-address any-
ipv4
set security policies from-zone INSIDE to-zone INTERNET policy DENY match application any
set security policies from-zone INSIDE to-zone INTERNET policy DENY then deny
set security policies from-zone INSIDE to-zone INTERNET policy DENY then log session-init

```

The third policy that needs to be created allows traffic from the servers in the DMZ access to the Management VLAN:

```

set security policies from-zone DMZ to-zone INSIDE policy ALLOW-MANAGEMENT match source-

```

```
address E-MAIL-SERVER
set security policies from-zone DMZ to-zone INSIDE policy ALLOW-MANAGEMENT match source-address WEB-SERVER
set security policies from-zone DMZ to-zone INSIDE policy ALLOW-MANAGEMENT match source-address PROXY-SERVER
set security policies from-zone DMZ to-zone INSIDE policy ALLOW-MANAGEMENT match destination-address MANAGEMENT-NETWORK
set security policies from-zone DMZ to-zone INSIDE policy ALLOW-MANAGEMENT match application junos-ntp
set security policies from-zone DMZ to-zone INSIDE policy ALLOW-MANAGEMENT match application junos-syslog
set security policies from-zone DMZ to-zone INSIDE policy ALLOW-MANAGEMENT match application junos-snmp-agentx
set security policies from-zone DMZ to-zone INSIDE policy ALLOW-MANAGEMENT then permit
```

The fourth policy allows the email server and proxy server access to relevant services on the Internet. The email server will only be allowed to access SMTP, whereas the proxy server can access HTTP, HTTPS, and FTP:

```
set security policies from-zone DMZ to-zone INTERNET policy ALLOW-E-MAIL match source-address E-MAIL-SERVER
set security policies from-zone DMZ to-zone INTERNET policy ALLOW-E-MAIL match destination-address any-ipv4
set security policies from-zone DMZ to-zone INTERNET policy ALLOW-E-MAIL match application junos-smtp
set security policies from-zone DMZ to-zone INTERNET policy ALLOW-E-MAIL then permit

set security policies from-zone DMZ to-zone INTERNET policy ALLOW-PROXY match source-address PROXY-SERVER
set security policies from-zone DMZ to-zone INTERNET policy ALLOW-PROXY match destination-address any-ipv4
set security policies from-zone DMZ to-zone INTERNET policy ALLOW-PROXY match application junos-http
set security policies from-zone DMZ to-zone INTERNET policy ALLOW-PROXY match application junos-https
set security policies from-zone DMZ to-zone INTERNET policy ALLOW-PROXY match application junos-ftp
set security policies from-zone DMZ to-zone INTERNET policy ALLOW-PROXY then permit
```

The fifth policy must allow Internet users access to the corporate web server via FTP, HTTP, and HTTPS, and access to the email server using SMTP. It is important to note that the branch office users will also need to access these servers. As the source address is set to any-ipv4, however, they will automatically be allowed access. But a rule allowing access to the proxy server should still be included:

```
set security policies from-zone INTERNET to-zone DMZ policy WEB-SERVER match source-address any-ipv4
set security policies from-zone INTERNET to-zone DMZ policy WEB-SERVER match destination-address WEB-SERVER
set security policies from-zone INTERNET to-zone DMZ policy WEB-SERVER match application junos-http
set security policies from-zone INTERNET to-zone DMZ policy WEB-SERVER match application junos-https
set security policies from-zone INTERNET to-zone DMZ policy WEB-SERVER match application junos-ftp
set security policies from-zone INTERNET to-zone DMZ policy WEB-SERVER then permit

set security policies from-zone INTERNET to-zone DMZ policy E-MAIL-SERVER match source-address any-
```

```

ipv4
set security policies from-zone INTERNET to-zone DMZ policy E-MAIL-SERVER match destination-
address E-MAIL-SERVER
set security policies from-zone INTERNET to-zone DMZ policy E-MAIL-SERVER match application junos-
smtp
set security policies from-zone INTERNET to-zone DMZ policy E-MAIL-SERVER then permit

set security policies from-zone INTERNET to-zone DMZ policy BRANCH-TO-PROXY match source-address MK-
BRANCH
set security policies from-zone INTERNET to-zone DMZ policy BRANCH-TO-PROXY match destination-
address PROXY-SERVER
set security policies from-zone INTERNET to-zone DMZ policy BRANCH-TO-PROXY match application junos-
http
set security policies from-zone INTERNET to-zone DMZ policy BRANCH-TO-PROXY match application junos-
https
set security policies from-zone INTERNET to-zone DMZ policy BRANCH-TO-PROXY match application junos-
ftp
set security policies from-zone INTERNET to-zone DMZ policy BRANCH-TO-PROXY then permit

```

The sixth and final policy is to allow access from the branch office into the corporate network:

```

set security policies from-zone INTERNET to-zone INSIDE policy BRANCH-OFFICE match source-
address MK-BRANCH
set security policies from-zone INTERNET to-zone INSIDE policy BRANCH-OFFICE match destination-
address CORP-VLAN
set security policies from-zone INTERNET to-zone INSIDE policy BRANCH-OFFICE match application any
set security policies from-zone INTERNET to-zone INSIDE policy BRANCH-OFFICE then permit

set security policies from-zone INTERNET to-zone INSIDE policy DENY match source-address any
set security policies from-zone INTERNET to-zone INSIDE policy DENY match destination-address any
set security policies from-zone INTERNET to-zone INSIDE policy DENY match application any
set security policies from-zone INTERNET to-zone INSIDE policy DENY then deny
set security policies from-zone INTERNET to-zone INSIDE policy DENY then log session-init

```

Once the configuration has been committed, best practice is to test whether the policies have worked successfully, but Internet access isn't allowed because there have been no NAT statements added to the configuration. This leads us nicely to *Chapter 3: NAT*.

Chapter 3

NAT

If the new SRX Series firewall were to be made live right now, you would have full reachability inside your own network. Users would be able to access the email server, web server, and even the proxy server, but that's as far as it would go. There would be no reachability to the global interconnected networks, and users wouldn't be able to access websites or to send or receive external email.

As you should have already figured out, the simple reason there is no connectivity outside the corporate network is because the servers and the clients are all using IP addresses from the RFC1918 address space.

TIP If ACME wanted to, it could configure servers in the DMZ with public IP addresses, but that would be a security risk, as attackers would know one of the address ranges in use inside the network. Apart from saving IPv4 addresses, NAT can help to obfuscate the addresses in use inside your LAN.

ACME, has created a firewall interface named DMZ, with three servers. Of these servers, two are accessed from outside the corporate LAN by clients on the Internet, and these are the web server and email server. The web server allows the public to view unencrypted pages via HTTP, encrypted pages via HTTPS, and allows authorized users to upload to the server via FTP. Traffic from Internet users will hit a public IP address on the INTERNET interface on the ASA that is translated to an RFC1918 address off of the DMZ interface.

The final server is a proxy server. This allows clients inside the network to access the Internet and as such it must accept requests from HTTP, HTTPS, and FTP, and pass on these requests to outside the network. As the traffic from the proxy server leaves the ASA, it must be translated to an outside address. Return traffic will be directed at the public IP address of the proxy server, which will then be translated back to the private IP address. Figure 3.1 shows the three servers in the DMZ and how each of these would NAT to the external addresses.

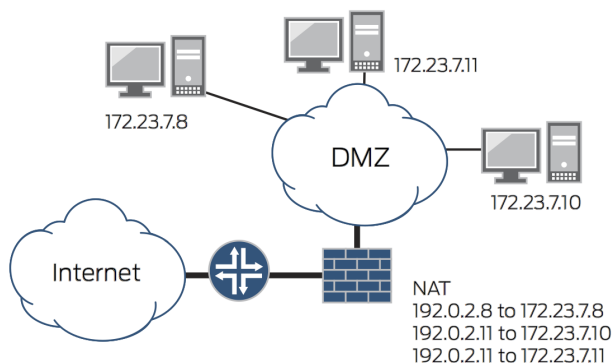


Figure 3.1 Servers in DMZ and NATs

You can see in Figure 3.1 that the web server has been given the IP address of 172.23.7.8 and this translates to the public address 192.0.2.8. The email server has been allocated the IP address 172.23.7.11, which is mapped to the public address of 192.0.2.11, and finally the proxy server has the address of 172.23.7.11, which the ASA translates to 192.0.2.11 as the traffic goes out to the Internet.

ASA NAT Configuration

As mentioned previously, the web server will allow connections to HTTP, HTTPS, and FTP, and the proxy server will need to access FTP, HTTP, and HTTPS. The mail server will only require SMTP, therefore the ASA has seven NATs configured just for these three servers. Each NAT statement deals with a separate protocol. The NATs on the ASA are configured as objects, with the host and NAT statement under each object. Accordingly, the ASA configuration is as follows:

```
object network E-MAIL_SERVER
host 172.23.7.8
nat (DMZ,INTERNET) static 192.0.2.8 service tcp smtp smtp

object network WEB-SERVER
host 172.23.7.11
nat (DMZ,INTERNET) static 192.0.2.11 service tcp www www

object network SECURE-WEB-SERVER
host 172.23.7.11
nat (DMZ,INTERNET) static 192.0.2.11 service tcp https https

object network FTP-SERVER
host 172.23.7.11
nat (DMZ,INTERNET) static 192.0.2.11 service udp ftp ftp

object network PROXY-SERVER-HTTP
host 172.23.7.10
nat (DMZ,INTERNET) static 192.0.2.10 service tcp www www

object network PROXY-SERVER-HTTPS
host 172.23.7.10
```

```
nat (DMZ,INTERNET) static 192.0.2.10 service tcp https https
object network PROXY-SERVER-FTP
host 172.23.7.10
nat (DMZ,INTERNET) static 192.0.2.10 service udp ftp ftp
```

The first seven NATs covered in this chapter are related to public access to the FTP, web and secure web server, and the mail server in order to allow the proxy server access to web servers, secure web servers, and FTP servers on the Internet. The next NAT is slightly different because having a proxy server in the LAN suggests that the proxy server will furnish Internet access requests on behalf of clients, but it provides restricted access and controls so that employees cannot just download copyrighted information or material that isn't safe for work. The type of NAT in use here is a one-to-one static NAT.

The Management VLAN doesn't access the Internet via the proxy, but instead has direct access for reasons that were discussed in Chapter 2. With the previous NATs, a single external IP address was mapped to a single internal IP address. The Management VLAN, however, is a /24 subnet, meaning that there are potentially 253 hosts that require 253 external NAT addresses. It is unlikely that the IT director would ever sign off on such a cost, therefore, in this case, the NAT would be a *NAT overload* where the entire subnet is set to NAT to a single outside address. Figure 3.2 provides a graphical representation of what the final NAT achieves.

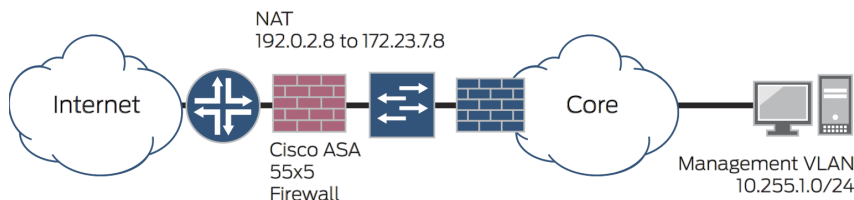


Figure 3.2 Servers in DMZ and NATs

Traffic from the Management VLAN, 10.255.1.0/24, reaches the ASA, which is then translated to the single address 192.0.2.9. This type of NAT uses the source address port to remember which traffic flow out of the ASA belongs to which client. For this action, the ASA uses the *xlate* table and in theory, as the source port is within the dynamic port address range, that single outside address should be able to handle 16,384 outgoing connection requests. As there are 253 potential clients, this allows 64 outgoing requests per user, which is more than enough.

The configuration on the ASA to allow this type of NAT is:

```
object network MAN-NAT
subnet 10.255.1.0 255.255.255.0
nat (INSIDE,INTERNET) static 192.0.2.9
```

SRX Series NAT Configuration

NAT statements on the SRX Series use a structure very similar to the firewall policy created in Chapter 2. The main difference between firewall policies and NAT statements is that the address-book is known as a pool. These statements are created by using the following syntax:

```
set security nat source pool <pool-name> address <IP-address>
```

If you recall from Chapter 2, you needed to state the *from* zone and *to* zone. In NAT configurations, zones are still referenced, but where the firewall policy might say something like:

```
set security policies from-zone INTERNET to-zone DMZ policy WEB-SERVER
```

In a static NAT statement, the rule set only states a *from* zone, rather like:

```
set security nat static <rule-name> from zone <zone-name>
```

Source and destination NATs require entry of the *from* and *to* zones, which segues to the next point – Junos allows for multiple types of NAT, and each has a different application. Table 3.1 helpfully lists these different types of NATs and their typical uses.

Table 3.1 NAT Types

NAT Type	Description
Source NAT	A Source NAT will translate the source address using only a pool of addresses, for example, a range of inside clients going through the SRX Series to web server, so the web server sees the source as a public IP address. The public addresses can be more than one address to allow thousands of clients to be translated.
Destination NAT	Destination NATs translate the destination address to a pool of addresses; for example, a client on the Internet accessing a web server inside your LAN. The request will be forwarded to one of several web servers, which will see the real source address, but the Internet client will never know the real IP address of the server to which its traffic was directed.
Double NAT	Double NATs are not used too often, as they translate both the source and destination addresses. They could be used where two companies merge and have overlapping address spaces, or where a client in one organization is trying to access servers in another.
Static NAT	Static NATs are for one-to-one mapping where the inside address is fixed directly to an outside address. Typically this would be where an outside client is accessing an inside resource such as an FTP server. Again, the client will never know the real address of the server.

ACME's Junos solution will use two NATs: the source and the static. The web server, including the secure web server and FTP server, and the mail server traffic, will be set to use static NATs. This is because static NATs translate on the

destination only, therefore these rules will be set as coming from the zone INTERNET. The match statements will reference the destination port in addition to the destination address and the then statement will tell the SRX Series which real IP address it will map to on the inside. First the rule set will be named FROM-INTERNET, then the *from* zone is set as INTERNET:

```
set security nat static rule-set FROM-INTERNET from zone INTERNET
```

To create the static NAT statements for only these servers, the following commands will be used. The first rule named WEB-SERVER will be for unencrypted web server traffic:

```
set security nat static rule-set FROM-INTERNET rule WEB-SERVER match destination-address 192.0.2.11/32
set security nat static rule-set FROM-INTERNET rule WEB-SERVER match destination-port 80
set security nat static rule-set FROM-INTERNET rule WEB-SERVER then static-nat prefix 172.23.7.11/32
set security nat static rule-set FROM-INTERNET rule WEB-SERVER then static-nat prefix mapped-port 80
```

The next rule sets a NAT for web server traffic on port 443. This rule is named SECURE-WEB-SERVER.

TIP If you keep the description relevant to what the rule is doing, it will help when it comes to troubleshooting:

```
set security nat static rule-set FROM-INTERNET rule SECURE-WEB-SERVER match destination-
address 192.0.2.11/32
set security nat static rule-set FROM-INTERNET rule SECURE-WEB-SERVER match destination-port 443
set security nat static rule-set FROM-INTERNET rule SECURE-WEB-SERVER then static-nat prefix 172.23.7.11/32
set security nat static rule-set FROM-INTERNET rule SECURE-WEB-SERVER then static-nat prefix mapped-port 443
```

Rule FTP-SERVER translates traffic destined for the FTP server, but to the same IP address as used for WEB-SERVER and SECURE-WEB-SERVER:

```
set security nat static rule-set FROM-INTERNET rule FTP-SERVER match destination-address 192.0.2.11/32
set security nat static rule-set FROM-INTERNET rule FTP-SERVER match destination-port 21
set security nat static rule-set FROM-INTERNET rule FTP-SERVER then static-nat prefix 172.23.7.11/32
set security nat static rule-set FROM-INTERNET rule FTP-SERVER then static-nat prefix mapped-port 21
```

The final static rule is for allowing outside access to the inside address of the mail server aptly named E-MAIL-SERVER:

```
set security nat static rule-set FROM-INTERNET rule E-MAIL-SERVER match destination-address 192.0.2.8/32
set security nat static rule-set FROM-INTERNET rule E-MAIL-SERVER match destination-port 25
set security nat static rule-set FROM-INTERNET rule E-MAIL-SERVER then static-nat prefix 172.23.7.8/32
set security nat static rule-set FROM-INTERNET rule E-MAIL-SERVER then static-nat prefix mapped-port 25
```

The NAT translation allowing the Management VLAN access to the Internet will be using a pool with a single address, but because real IP addresses are a whole subnet of addresses, a source NAT has to be used. This NAT will be using Port Address Translation (PAT). On the other hand, the proxy server is still using a pool with a single address, but because you are specifying the source address, you cannot set this NAT to be static, so a source NAT is being used instead. You do not want any PAT used for this NAT, therefore when creating the pool named PROXY-SERVER, the option port no-translation should be specified:

```
set security nat source pool MAN-OUTSIDE address 192.0.2.9/32
set security nat source pool PROXY-SERVER address 192.0.2.10/32
set security nat source pool PROXY-SERVER port no-translation
```

There are two rule sets created for the source NATs. The first rule set is from the zone `INSIDE` to the zone `INTERNET` and as such will be given the name `FROM-INSIDE`. The second rule set will be named `FROM-DMZ` and will be from the zone `DMZ` to the zone `INTERNET`. Unlike static NATs, you do need to mention the `from` zone and the `to` zone:

```
set security nat source rule-set FROM-INSIDE from zone INSIDE
set security nat source rule-set FROM-INSIDE to zone INTERNET

set security nat source rule-set FROM-DMZ from zone DMZ
set security nat source rule-set FROM-DMZ to zone INTERNET
```

Once the pools have been created and the rule sets have specified the `from` and `to` zones, the NAT source statements can be created. The first is for the Management VLAN. In this statement, the destination has to be set and as this is for any traffic, the address `0.0.0.0/0` is specified:

```
set security nat source rule-set FROM-INSIDE rule MAN-NAT match source-address 10.255.1.0/24
set security nat source rule-set FROM-INSIDE rule MAN-NAT match destination-address 0.0.0.0/0
set security nat source rule-set FROM-INSIDE rule MAN-NAT then source-nat pool MAN-OUTSIDE
```

The next rule set tells the SRX Series to translate packets from the proxy server address of `172.23.7.10` to the pool `PROXY-SERVER`, which has the address `192.0.2.10`. The difference here is that you do not want the SRX Series to translate all packets, just those related to FTP, HTTP, and HTTPS, so therefore three rules are created, one for each port. The rule for HTTP traffic is created first:

```
set security nat source rule-set FROM-DMZ rule PROXY-SERVER-WEB match source-address 172.23.7.10/32
set security nat source rule-set FROM-DMZ rule PROXY-SERVER-WEB match destination-address 0.0.0.0/0
set security nat source rule-set FROM-DMZ rule PROXY-SERVER-WEB match destination-port 80
set security nat source rule-set FROM-DMZ rule PROXY-SERVER-WEB then source-nat pool PROXY-SERVER
```

The second rule is for HTTPS or secure web server traffic:

```
set security nat source rule-set FROM-DMZ rule PROXY-SERVER-SECURE match source-address 172.23.7.10/32
set security nat source rule-set FROM-DMZ rule PROXY-SERVER-SECURE match destination-address 0.0.0.0/0
set security nat source rule-set FROM-DMZ rule PROXY-SERVER-SECURE match destination-port 443
set security nat source rule-set FROM-DMZ rule PROXY-SERVER-SECURE then source-nat pool PROXY-SERVER
```

Finally, the rule for FTP server traffic is created:

```
set security nat source rule-set FROM-DMZ rule PROXY-SERVER-FTP match source-address 172.23.7.10/32
set security nat source rule-set FROM-DMZ rule PROXY-SERVER-FTP match destination-address 0.0.0.0/0
set security nat source rule-set FROM-DMZ rule PROXY-SERVER-FTP match destination-port 21
set security nat source rule-set FROM-DMZ rule PROXY-SERVER-FTP then source-nat pool PROXY-SERVER
```

```
set security nat proxy-arp interface ge-0/0/0.0 address 192.0.2.8/32 to 192.0.2.11/32
```

Testing the NAT Configuration

After applying the customary commit, this configuration can now be tested. As there are really no web servers or mail servers behind this device in the lab, it's a little tricky to perform a proper test. What you can do, however, is open a Telnet session from another router but specify the port as that of the mail server or web server:

```
TerminalServer#Telnet 192.0.2.8 25
```

Trying 192.0.2.8, 25 ...

Once completed, the next useful command allows you to see if the rule has been hit:

show security nat static rule <rule-name>

So if this command was run for the rule E-MAIL-SERVER, there should be evidence of the SRX Series attempting to translate the packet to the inside address. Let's check:

root@ACME-INTERNET-FW> show security nat static rule E-MAIL-SERVER

```
Static NAT rule: E-MAIL-SERVER      Rule-set: FROM-INTERNET
Rule-Id                            : 4
Rule position                       : 4
From zone                          : INTERNET
Destination addresses               : 192.0.2.8
Destination ports                   : 25 - 25
Host addresses                      : 172.23.7.8
Host ports                          : 25 - 25
Netmask                             : 32
Host routing-instance              : N/A
Translation hits                    : 8
  Successful sessions               : 0
Number of sessions                  : 0
```

As you can see, there were eight attempts before the device timed out. There were zero successful sessions because there was no device behind the SRX Series (in the lab) to respond to the session request. Testing the source NAT rules is a little trickier because, again, there are no devices on the inside or outside yet. You can still check the configuration by running the show security nat source summary command:

root@ACME-INTERNET-FW> show security nat source summary
Total port number usage for port translation pool: 64512
Maximum port number for port translation pool: 8388608

Total pools: 2

Pool	Address	Routing	PAT	Total
Name	Range	Instance		Address
MAN-OUTSIDE	192.0.2.9-192.0.2.9	default	yes	1
PROXY-SERVER	192.0.2.10-192.0.2.10	default	no	1

Total rules: 4

Rule name	Rule set	From	To	Action
MAN-NAT	FROM-INSIDE	INSIDE	INTERNET	MAN-OUTSIDE
PROXY-SERVER-WEB	FROM-DMZ	DMZ	INTERNET	PROXY-SERVER
PROXY-SERVER-SECURE	FROM-DMZ	DMZ	INTERNET	PROXY-SERVER
PROXY-SERVER-FTP	FROM-DMZ	DMZ	INTERNET	PROXY-SERVER

You can see that the command lists the pools, the rule set, and the rule name. The Action column lists the pool that NAT will use when it is hit.

Now that the NATs have been added, users should be able to access the Internet, send and receive emails, and upload files, and in return, clients on the Internet should be able to access the corporate website, send emails, and upload files, while the users in the Management VLAN can do whatever they like. But aren't we forgetting something? What about the users in the branch office? They require a site-to-site VPN connection from the branch office into the headquarters, and this is exactly what Chapter 4 covers.

Chapter 4

Site-to-Site VPN

Site-to-site VPNs are commonplace in today's interconnected world, whether they are connecting branch offices or third parties who provide outsourced services. As the Internet has matured, it has provided a stable and cost-effective medium to interconnect locations, something that in the past was reserved for large enterprises who could afford private WANs. In this book's use case, ACME is using an IPsec VPN to connect a branch office to their headquarters as illustrated in Figure 4.1.

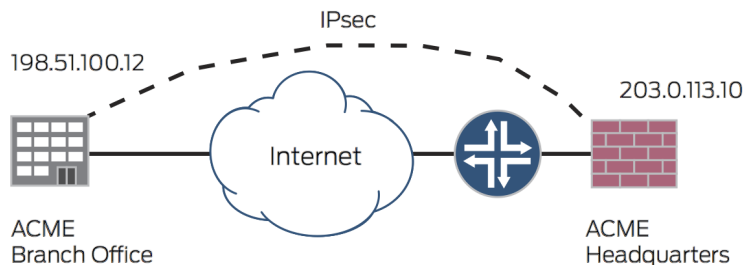


Figure 4.1 Site-to-Site VPN

While different networking vendors use different terminologies for the components required to form an IPsec VPN, fundamentally they are the same: IKE Phase 1 proposal, IPsec Phase 2 proposal, local and remote networks, and an interface on which the VPN terminates. The ASA configuration is a policy-based VPN, but that loses its value on the SRX Series, so let's configure the more commonly used tunnel or route-based VPN. You create a logical tunnel interface that the IPsec tunnel is bound to, and use the routing table rather than firewall policy to determine the path of the VPN traffic.

ASA Configuration

The ASA configuration is not as hierarchical and structured as the SRX Series configuration, so when translating the configuration, you need to jump around a little through different sections of the configuration. Below is the full configuration as it would appear on the ASA.

TIP In the following configuration steps the relevant ASA commands are shown with the corresponding SRX Series commands:

```
access-list SITE-T0-SITE-VPN_acl extended permit ip 10.0.0.0 255.0.0.0 10.200.0.0 255.255.240.0
access-list SITE-T0-SITE-VPN_acl extended permit ip 172.23.7.0 255.255.255.0 10.200.0.0 255.255.240.0

crypto ikev1 enable INTERNET
crypto ikev1 policy 10
authentication pre-share
encryption aes
hash sha
group 5
lifetime 1080

crypto ipsec ikev1 transform-set ACMEVPN-TSET esp-aes esp-sha-hmac
crypto ipsec security-association lifetime seconds 3600
crypto ipsec security-association pmtu-aging infinite

crypto map ACMEVPN interface OUTSIDE
crypto map ACMEVPN 10 match address SITE-T0-SITE-VPN_acl
crypto map ACMEVPN 10 set pfs
crypto map ACMEVPN 10 set peer 198.51.100.12
crypto map ACMEVPN 10 set ikev1 transform-set ACMEVPN-TSET

group-policy ACMEVPN internal
group-policy ACMEVPN attributes
vpn-tunnel-protocol ikev1
pfs enable

tunnel-group 198.51.100.12 type ipsec-l2l
tunnel-group 198.51.100.12 general-attributes
default-group-policy ACMEVPN
tunnel-group 198.51.100.12 ipsec-attributes
ikev1 pre-shared-key 5up3r53c43t
```

SRX Series Junos Configuration

The Junos OS is very hierarchical, and VPN configuration follows logic that is similar to a lot of other features. The phase 1 configuration contains three components: the IKE proposal, the IKE policy, and the IKE gateway.

ASA

```
crypto ikev1 enable INTERNET
crypto ikev1 policy 109
authentication pre-share
encryption aes
hash sha
group 5
lifetime 1080
```

SRX

The following commands define the IKE proposal and this is where you define the authentication method that will be referenced in the IKE policy. In terms of the ASA configuration, the following relates directly to the `crypto ike policy`:

```
set security ike proposal ACME-IKE-PROP authentication-method pre-shared-keys
set security ike proposal ACME-IKE-PROP dh-group group5
set security ike proposal ACME-IKE-PROP authentication-algorithm sha1
set security ike proposal ACME-IKE-PROP encryption-algorithm aes-128-cbc
```

Next you define the IKE policy, within the policy in which you defined the previously configured IKE proposal. While you are only using one proposal in this configuration, multiple proposals can be referenced:

SRX

```
set security ike policy ACME-IKE-POLICY mode main
set security ike policy ACME-IKE-POLICY proposals ACME-IKE-PROP
set security ike policy ACME-IKE-POLICY pre-shared-key ascii-text
```

The IKE gateway configuration completes the phase 1 configuration, now let's tie together the previously created policy, the remote gateway IP address, and the external interface of the SRX Series:

ASA

```
crypto map ACMEVPN 10 set peer 198.51.100.12

crypto map ACMEVPN interface OUTSIDE
```

SRX

```
set security ike gateway ACME-IKE-GW ike-policy ACME-IKE-POLICY
set security ike gateway ACME-IKE-GW address 198.51.100.12
set security ike gateway ACME-IKE-GW external-interface ge-0/0/0.0
```

ASA

```
crypto ipsec ikev1 transform-set ACMEVPN-TSET esp-aes esp-sha-hmac
crypto ipsec security-association lifetime seconds 3600
crypto ipsec security-association pmtu-aging infinite
```

SRX

As with the phase 1 configuration for IKE, the IPsec configuration is made up from a proposal, a policy, and then finally the VPN:

```
set security ipsec proposal ACME-IPSEC-PROP protocol esp
set security ipsec proposal ACME-IPSEC-PROP authentication-algorithm hmac-sha1-96
set security ipsec proposal ACME-IPSEC-PROP encryption-algorithm aes-128-cbc
set security ipsec proposal ACME-IPSEC-PROP lifetime-seconds 3600
set security ipsec proposal ACME-IPSEC-PROP lifetime-kilobytes 4608000
set security ipsec policy ACME-IPSEC-POLICY perfect-forward-secrecy keys group2
set security ipsec policy ACME-IPSEC-POLICY proposals ACME-IPSEC-PROP
```

Now that the encryption and authentication mechanisms have been defined, let's create the logical tunnel interface for the VPN to be bound to. The limits on the number of SRX Series interfaces you can have differs between SRX Series models, but given the encryption overhead of the VPNs, the SRX Series will run out of resources before it runs out of logical interfaces.

Configuring the tunnel interface is just like any other interface on the SRX Series and follows the same hierarchical command structure used earlier in this book.

At the moment, the configuration allows a single branch VPN access into the headquarters, therefore this is technically a point-to-point VPN. That said, there is a possibility that further branches may require VPN access at a later date, and as such, the headquarters SRX Series will become the hub in a hub-and-spoke environment. Therefore, in addition to creating the interface `st0` and assigning the IP address to this interface, it would also be a good idea to specify the option `multi-point`, and to allow multiple VPN tunnels into this device:

```
set interfaces st0 unit 0 multipoint
set interfaces st0 unit 0 family inet address 10.200.1.33/28
```

Now the final stage is to take our IKE, IPsec policies, and tunnel interface, and combine them to build the VPN tunnel. You'll note that there are two groups of almost identical commands, the difference being the remote networks.

SRX

First bind the VPN to the logical tunnel interface:

```
set security ipsec vpn ACME-VPN bind-interface st0.0
```

Then define the previously created IKE gateway configuration:

```
set security ipsec vpn ACME-VPN ike gateway ACME-IKE-GW
```

Next you define the local and remote networks. First how you did so with the ASA and then what to do on the SRX Series:

ASA

```
access-list SITE-T0-SITE-VPN_acl extended permit ip 10.0.0.0 255.0.0.0 10.200.0.0 255.255.240.0
access-list SITE-T0-SITE-VPN_acl extended permit ip 172.23.7.0 255.255.255.0 10.200.0.0 255.255.240.0
```

SRX

```
set security ipsec vpn ACME-VPN ike proxy-identity local 10.200.0.0/20
set security ipsec vpn ACME-VPN ike proxy-identity remote 10.0.0.0/9
```

Now define the IPsec policy previously created and set the tunnel to establish immediately, by default, whenever some form of traffic is initiated by the VPN:

```
set security ipsec vpn ACME-VPN ike ipsec-policy ACME-IPSEC-POLICY
set security ipsec vpn ACME-VPN establish-tunnels immediately

set security ipsec vpn ACME-VPN2 bind-interface st0.0
set security ipsec vpn ACME-VPN2 ike gateway ACME-IKE-GW
set security ipsec vpn ACME-VPN2 ike proxy-identity local 10.200.0.0/20
set security ipsec vpn ACME-VPN2 ike proxy-identity remote 172.23.7.0/24
set security ipsec vpn ACME-VPN2 ike ipsec-policy ACME-IPSEC-POLICY
set security ipsec vpn ACME-VPN2 establish-tunnels immediately
```

Command for command, the VPN configuration is now migrated from the ASA to the SRX Series and the VPN would be established at ACME. However, as previously mentioned, the ASA was using a policy-based VPN and the SRX Series is now using a route-based VPN. As the name suggests, before you commit the configuration you need to create static routes on the SRX Series for the remote networks:

```
set static route 10.0.0.0/9 next-hop 10.200.1.33
set static route 172.23.7.0/24 next-hop 10.200.1.33
```

Chapter 5

Device Management

If the SRX Series were migrated at this point, there would be full connectivity. The VPN would be up to allow the branch office to connect to headquarters, and users in the headquarters would be able to surf the Internet, and so on. However, if this device were to go live right now, it stands to face an increased risk of being compromised from an attack, and the reason for this is because the device hasn't been hardened.

In an era where IT systems and services are suffering from an increased number of attacks, it is essential that all network devices are configured to defend themselves against unauthorized access. This need is increased when the network device is Internet-facing like this SRX Series will be.

On this SRX Series, restrictions must be put in place to prevent anyone from outside ACME from being able to administer this device. In addition, access from inside the organization should be restricted to the management subnets, for internal ACME reasons. Figure 5.1 shows a sample of some of the VLANs inside ACME.

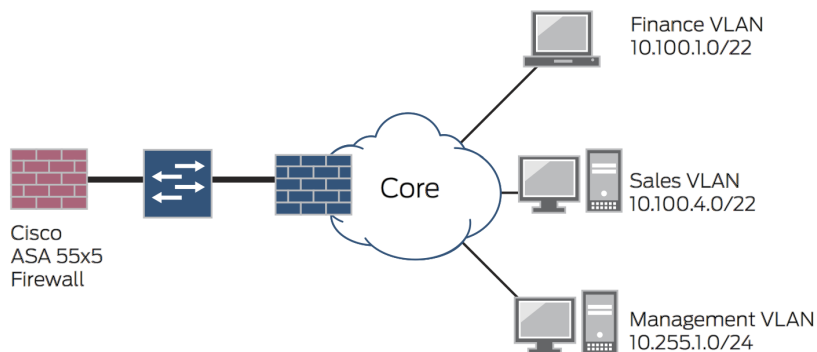


Figure 5.1 ACME HQ Subnets

So the SRX Series needs to be configured to only allow access from the Management VLAN, and this access should be restricted to SSH and HTTPS for configuration purposes, and to SNMP for monitoring. Access from Telnet and unencrypted HTTP must be denied. Let's harden the SRX Series using your available ASA knowledge.

Restricting ICMP Echo Requests

The first option that was configured on the ASA is to restrict Internet Control Message Protocol (ICMP) echo requests, or pings, from sources on the public Internet. The ASA has a mechanism that is quite simple to configure in which the administrator can specify which IP address or addresses are allowed to ping the device and on which interface. In this case, this ASA has been configured to allow pings from only the branch office and to allow pings from hosts that are inside the network:

```
icmp permit 198.51.100.12 255.255.255.255 echo INTERNET
icmp permit 198.51.100.12 255.255.255.255 echo-reply INTERNET
icmp permit 198.51.100.12 255.255.255.255 time-exceeded INTERNET
icmp permit 198.51.100.12 255.255.255.255 unreachable INTERNET
icmp permit 10.0.0.0 255.0.0.0 echo INSIDE
icmp permit 10.0.0.0 255.0.0.0 echo-reply INSIDE
icmp permit 10.0.0.0 255.0.0.0 time-exceeded INSIDE
icmp permit 10.0.0.0 255.0.0.0 unreachable INSIDE
```

As there is an implicit deny, no other sources from the public Internet will receive a reply if they attempt to ping this device. The SRX Series, on the other hand, is slightly different. First, by default, the SRX Series is configured not to accept ICMP requests. This can be allowed by entering the following command:

```
set security zones security-zone RTR-FW-UPLINK interfaces ge-0/0/0.0 host-inbound-traffic system-
services ping
```

Once ICMP is allowed, traffic must then be restricted to only the required IP addresses. The firewall policy that is currently in place only restricts traffic passing through the SRX Series. It does not affect traffic directed at the SRX Series. This is because the firewall policy controls traffic entering the data plane, whereas traffic directed at the SRX Series enters the control plane, also known as the RE, or Routing Engine. This means traffic needs to be restricted from entering the Routing Engine and this is done in a similar way to the firewall policy and NAT statements, through the use of a firewall filter.

The filter itself will be doing more than just allowing and denying ICMP, therefore the filter will be given the name of `PROTECT-SRX`, after which a term named `ICMP` is created with specifies to both the address of the branch office and a summarized address of the inside subnets and the protocol, which is ICMP. After this the then action of accept is specified:

```
set firewall family inet filter PROTECT-SRX term ICMP from protocol icmp
set firewall family inet filter PROTECT-SRX term ICMP from source-address 198.51.100.12/32;
set firewall family inet filter PROTECT-SRX term ICMP from source-address 10.0.0.0/8
set firewall family inet filter PROTECT-SRX term ICMP then accept
```

As traffic is implicitly denied, there is no need to add a rule denying ICMP traffic. If you wanted to log any denials and count the number of attempts that were made, then a rule could be added with `discard`, followed by `log` and `count` instead of `accept`, however, with the number of clients on the Internet, the amount of random ping attempts could be quite high. You can do that in your lab if you wish, but for ACME it's better to not count and log attempts so as not to fill the log file with entries.

The question you might have is: *how to apply this filter to the SRX Series?* The SRX Series routing engine isn't exactly in its own zone and you can't apply this filter to the interface `ge-0/0/2.0` as, again, this would affect traffic traversing the data plane. The answer is quite surprising. When the SRX Series has the default configuration, there are multiple loopback interfaces created for no apparent reason. If the filter is applied to interface `lo0.0`, the SRX Series will use this filter to protect the RE. The command used to apply the filter is therefore:

```
set interface lo0.0 family inet filter input PROTECT-SRX
```

CAUTION At this point it would be a very bad idea to issue a commit to this SRX Series, primarily because the filter does not specify other allowed traffic, such as SSH, and you could quite easily lock yourself out of the device, therefore the commit will be issued at the end of this chapter.

Securing Web GUI and CLI traffic

Some network engineers do not like to use the Web GUI. Some do, because it's useful for monitoring, but no matter what the reason, this ASA currently has HTTPS enabled on the INSIDE interface and only allows traffic from the Management VLAN. This is done with the following configuration:

```
http server enable
http 10.255.1.0 255.255.255.0 INSIDE
```

The SRX Series by default allows management via HTTP and HTTPS, but only from the default VLAN. The default configuration is shown here:

```
[edit system services]
root@ACME-INTERNET-FW# show
ssh;
Telnet;
xnm-clear-text;
web-management {
    http {
        interface vlan.0;
    }
}
```

```
https {  
    system-generated-certificate;  
    interface vlan.0;
```

Therefore, this configuration needs to be changed to delete HTTP management completely, to delete vlan.0 from the HTTPS configuration, and to add the interface in the INSIDE zone, which is interface ge-0/0/2.0, to this configuration instead:

```
delete system services web-management http  
delete system services web-management https interface vlan.0  
set system services web-management https interface ge-0/0/2
```

As most engineers will be configuring the device via CLI, SSH is enabled and as it is insecure, Telnet is disabled. This has already been removed from the ASA, therefore the only commands that remain in place that need to be converted to the SRX Series are related to SSH. These commands specify that the SSH version should be Version 2 and that only the Management VLAN may connect to this device for management purposes:

```
ssh 10.255.1.0 255.255.255.0 INSIDE  
ssh version 2
```

As shown in the default configuration, Telnet is enabled by default and this should be deleted, after which the SSH version should be set to 2. It is also a good idea to prohibit the root user from being able to access this device via SSH due to the damage it could cause in the wrong hands, therefore, while this isn't an option on the ASA, it will be added to the SRX Series in any case:

```
delete system services Telnet  
set system services ssh protocol-version v2  
set system services ssh root-login deny
```

Where the ASA specified that SSH should only be allowed on the INSIDE interface, technically this has already been done on the SRX Series. When the security zones were configured, the zone trust was renamed as the zone INSIDE. When this was done, the following options were moved across to the zone INSIDE, too:

```
root@ACME-INTERNET-FW> ...curity zones security-zone INSIDE  
host-inbound-traffic {  
    system-services {  
        all;  
    }  
    protocols {  
        all;  
    }  
}
```

This option allows all traffic directed at the SRX Series to be accepted by the routing engine unless the firewall filter assigned to interface lo0.0 denies the traffic. At the same time, there is no such option set on the INTERNET and DMZ interfaces, so not even a ping is allowed by default, and that's why it was necessary earlier to set the host-inbound-traffic system-services ping option on the ge-0/0/0.0 interface.

Restricting which IP addresses can access this device must be done via the same firewall filter that was created earlier. This simply means adding another term called `DEVICE-MANAGE` to the filter `PROTECT-SRX`:

```
set firewall family inet filter PROTECT-SRX term DEVICE-MANAGE from source-address 10.255.1.0/24
set firewall family inet filter PROTECT-SRX term DEVICE-MANAGE from protocol tcp
set firewall family inet filter PROTECT-SRX term DEVICE-MANAGE from destination-port ssh
set firewall family inet filter PROTECT-SRX term DEVICE-MANAGE from destination-port https
set firewall family inet filter PROTECT-SRX term DEVICE-MANAGE then accept
```

Let's assume the ACME IT director would like a list of any unauthorized addresses that have tried to access this device and a count of how many attempts have been made. Therefore, another term with the name `DENY-MANAGE` will be created, which encompasses HTTP, HTTPS, Telnet, and SSH. As the source address is not specified, this filter will apply to all addresses not covered in previous terms:

```
set firewall family inet filter PROTECT-SRX term DENY-MANAGE from protocol tcp
set firewall family inet filter PROTECT-SRX term DENY-MANAGE from destination-port ssh
set firewall family inet filter PROTECT-SRX term DENY-MANAGE from destination-port https
set firewall family inet filter PROTECT-SRX term DENY-MANAGE from destination-port http
set firewall family inet filter PROTECT-SRX term DENY-MANAGE from destination-port Telnet
set firewall family inet filter PROTECT-SRX term DENY-MANAGE then count ACCESS-DENIED
set firewall family inet filter PROTECT-SRX term DENY-MANAGE then log
set firewall family inet filter PROTECT-SRX term DENY-MANAGE then reject
```

At the end of this term, you can see the action of `reject` is used along with a `log` and a count, which will send the counts to a counter called `ACCESS-DENIED`. The alternative option to `reject` is `discard`. The difference between these options is that `discard` silently ignores the packet whereas `reject` will send a response back to the client stating that this request wasn't allowed.

One issue with using `reject` is that when an attacker tries to access an address they would not be allowed access, but because they get a reply they know that the IP address they attempted to access was in use, and as a result know they can attack it. This doesn't matter as much in our case as the only interface SSH, HTTP, HTTPS, and Telnet traffic can come in on is `ge-0/0/2.0`.

There is one major issue that must be addressed. This filter has an implicit deny. This means that any traffic that isn't specifically allowed by this filter will be denied, which includes SNMP traffic or dynamic routing protocols that may be in use. This means that the filter should end with a term allowing all other traffic access to the device. If no source, destination, or protocol is set, then this term will match all traffic not mentioned in previous terms:

```
set firewall family inet filter PROTECT-SRX term EVERYTHING-ELSE then accept
```

It should now be safe to commit this configuration to the SRX Series.

Can a Firewall Tell the Time?

Network devices, whether they are switches, firewalls, or routers, use log files to record important events such as a downed interface or denied access attempts. If there were an attack on the network, the logs could be correlated in order to build a picture of the attack. But that's only possible if the time on all network devices is exact. This is what the Network Time Protocol (NTP) does.

ACME uses an internal NTP server as an accurate time source for their network devices. These servers are located on the Management VLAN. They also have authentication enabled on NTP clients where NTP clients will authenticate the NTP server using a MD5 hash, which is set to T1m3-F1135.

NTP servers and clients can be configured with multiple authentication strings by assigning a key number to each string. For example, one string can be set with a key of 1, whereas a second string can have a key of 2. This allows devices to migrate from one string to another without any period where authentication either fails or in which there is no authentication. In ACME's case, the string has been set with the key of 123 and the ASA configuration below reflects all of this information:

```
ntp authentication-key 123 md5 T1m3-F1135
ntp authenticate
ntp trusted-key 123
ntp server 10.255.1.25 key 123 source INSIDE prefer
```

The SRX Series configuration is fairly straightforward and shares some similarities with the ASA. With the SRX Series, however, there is no need to tell the device to use authentication, as the SRX Series will automatically do so as soon as the key is specified after the server IP address. There is a need, however, to tell the SRX Series to use the NTP server as soon as the device boots in order to immediately eliminate any time drift, therefore the option `boot-server` is added to the NTP configuration:

```
set system ntp authentication-key 123 type md5 value T1m3-F1135
set system ntp server 10.255.1.25 key 123
set system ntp trusted-key 123
set system ntp boot-server 10.255.1.25
```

Login Authentication

Although it is possible to administer the device without a username and password on the ASA, it is certainly not ideal, as this is a major security risk. With the SRX Series, you are forced to enter a password for the root user when you perform the initial configuration on the SRX Series. Although the SRX Series has the option of a root user, that doesn't mean it's a good idea to continue to use this user account, primarily because this account gives an administrator full access to the underlying operating system.

In either case, as part of corporate security requirements, ACME requires a *local account* to be created. This account is named *ACMEadmin* and is set with a complex password of *\$\$B1lly4ll*, and in the case of the ASA had Cisco's *privilege 15* rights, therefore the ASA had the following configuration added:

```
username ACMEadmin password $$B1lly4ll privilege 15
```

To duplicate the creation of a local account on the SRX Series, the following command would be used:

```
set system login user ACMEadmin class super-user plain-text-password
```

After pressing the Enter key, the Junos OS prompts the administrator to enter the password and then re-enter the password. Unlike the ASA, the password in Junos OS is hidden as it is typed when the username is created.

Using this local administrator account, however, does have some drawbacks. The first drawback is keeping track of who has made what changes. As you will see later, when a commit is performed within the Junos OS the messages sent to logs are quite verbose, and the user that performed the commit is included within the messages. The better option would be to have each user use their own username, which brings us to the second drawback.

If the network team is of a reasonable size, and if the organization has a large number of devices, adding a username when someone joins the team or removing a username from devices when an engineer leaves can be quite time intensive. The most sensible option would be to leave a single local administrator account and use an Access Control Server (ACS) to authenticate administrator accounts from a central location.

With an ACS server, the username and password only need to be set on that server and the network devices connect to the ACS server to perform authentication. The local user account is there as a backup in case the ACS server is inaccessible. There are two types of ACS servers: RADIUS and TACACS+. ACME use a TACACS+ server with the IP address of 10.155.1.30. The ACS server needs to be configured with a *secret*, which the network device will use to prove to the ACS server it is an authorized device. The secret on the ASA is set to *5up3r53c43t*:

```
aaa-server TACACS-SERVER protocol tacacs+  
aaa-server TACACS-SERVER host 10.255.1.30 5up3r53c43t
```

Typically, the secret should be set differently for each device, however as the ASA is being replaced by the SRX, and as such the host name and IP address remains the same, the secret on the SRX should be set to the same secret in use on the ASA. This command sets this on the SRX along with the IP address of the ACS server:

```
set system tacplus-server 10.255.1.30 secret 5up3r53c43t
```

After the TACACS+ server has been configured, the device needs to be told to use that AAA server. With the ASA there were multiple components that need to be authenticated, SSH, the WebGUI, and the enable password. This means three statements need to be added to tell the ASA to use the ACS server for all of these connection options:

```
aaa authentication ssh console TACACS-SERVER LOCAL
aaa authentication http console TACACS-SERVER LOCAL
aaa authentication enable console TACACS-SERVER LOCAL
```

After authentication – which verifies that someone is who they say they are – comes authorization, which determines what the user can do, assuming the authentication check confirms that the user is permitted to access that device. The authorization options are set on the ACS server and are not covered in this book, but the commands to tell the ASA to use AAA authorization are below:

```
aaa authorization command TACACS-SERVER LOCAL
aaa authorization exec authentication-server
```

To tell the Junos OS to authenticate using the ACS server, you need to set the authentication order, otherwise Junos OS will continue to authenticate using the local username. The keyword of *tacplus* tells Junos OS to use the TACACS+ server first, then the keyword of *password* tells Junos OS to use the local usernames should the ACS server be unreachable or if it times out. Should authentication fail, the Junos OS does not fall back to the local user name, which is similar to the way the ASA handles authentication:

```
set system authentication order [tacplus password]
```

Authorization on the SRX Series is slightly different. Junos OS uses *classes* that are associated with the username. These classes are created within Junos OS. By default, three classes are created: *super-user*, *operator*, and *read-only*.

Therefore, with the SRX Series, the ACS only needs to tell the SRX Series which of these classes any authenticated user belongs to, and this is done by the use of templates:

```
set system login user remote-super-users full-name "Template for super-users" uid 2012 class super-user
set system login user remote-operator full-name "Template for Operators" uid 2013 class super-user
set system login user remote-read-only full-name "Template for read-only" uid 2014 class read-only
```

Finally, the TACACS+ server needs to be told to tell Junos OS which class the authenticated user is a part of. Typically, groups would be created on the ACS server, users would be assigned to these groups, then a service setting is applied to each group. As an example, the service setting for the class *super-user*, based on the templates that were created in the previous step, would be as follows:

```
service = junos-exec {
    local-user-name = remote-super-users
```

MORE? While this book can't provide all the information on how to configure the ACS server for Junos OS authorization, the following knowledge-based article may prove useful for engineers wishing to study this topic in a little more detail: <https://kb.juniper.net/InfoCenter/index?page=content&id=KB17269&actp=search>.

Monitoring and Managing via SNMP

The final hardening that needs to be configured relates to monitoring the device, because engineers need to inspect logs if they are trying to diagnose an issue. And if your network is large enough, the network operations center team needs to know when the device has failed or when an interface goes down.

The first type of monitoring is known as syslog and there are eight levels of numerical syslog logging, from 0 for emergencies up to 7 for debugging. The ASA being replaced was configured to send logs of Level 6, also known as *informational* or *higher*, and the ASA has been configured to send these syslog messages to the IP address 10.255.1.20 via the INSIDE interface:

```
logging host INSIDE 10.255.1.20
logging buffered informational
```

When configuring the SRX Series, the syslog level needs to be specified with the host IP address. In addition, the facilities that are being monitored, such as ports or configuration changes, need to be specified. It's possible to specify different facilities for different log servers, but ACME wants all facilities to go to the same syslog server, so here, the keyword *any* is specified after the host IP address:

```
set system syslog host 10.255.1.20 any info
```

The second type of monitoring is SNMP monitoring. SNMP doesn't collect syslog messages but instead monitors the device, sending pings to ensure the device is up, sending *gets* to find what the CPU or interface usage percentage is, and receiving traps, which the device will send to the SNMP server when, say, an interface goes down, or after a reboot.

The SNMP server in use at ACME is reachable via the IP address 10.255.1.40. There are two uses of the community string "NOT-PUBLIC" in the following configuration: the first use relates to getting requests from the SNMP server, whereas the second use is for traps and this command also includes the server IP. The version of SNMP the server is configured to use is 2c. The ASA was configured to send all traps to the server. Finally, the ASA has been configured with a location and contact details. All of this information is reflected here in the ASA:

```
snmp-server community NOT-PUBLIC
snmp-server host INSIDE 10.255.1.40 community NOT-PUBLIC version 2c
snmp-server location HQ
snmp-server contact admin@ACME.com
snmp-server enable traps all
```

Configuring SNMP on the SRX Series is slightly more involved than it is on the ASA. The first thing that needs to be done is to specify a *trap-group*. This groups all of the commands related to sending traps to the server under one hierarchy, and it also allows the configuration of multiple groups so that some traps can go to one server and other traps can go to another.

In this case, the trap group will be given the name `SNMP` and the first commands that will be entered will set the version to 2c and the SNMP server IP address and, as there are multiple addresses configured on this server, the source address is also specified. This is set to the IP address of interface `ge-0/0/2.0`:

```
set snmp trap-group SNMP version v2
set snmp trap-group SNMP targets 10.155.1.40
set snmp trap-options source-address 10.100.1.1
```

Finally, you need to tell Junos OS which traps to send to the SNMP server. The ASA was set to send all traps, and the SRX Series doesn't allow this option, but you can clearly see there are several other options for your consideration in the following output:

```
[edit]
root@ACME-INTERNET-FW# set snmp trap-group SNMP categories ?
Possible completions:
+ apply-groups          Groups from which to inherit configuration data
+ apply-groups-except  Don't inherit configuration data from these groups
  authentication        Authentication failures
  chassis               Chassis or environment notifications
  chassis-cluster       Clustering notifications
  configuration          Configuration notifications
  link                  Link up-down transitions
> otn-alarms            OTN alarm trap subcategories
  remote-operations     Remote operations
  rmon-alarm            RMON rising and falling alarms
  routing               Routing protocol notifications
  services              Services notifications
> sonet-alarms          SONET alarm trap subcategories
  startup               System warm and cold starts
  vrrp-events           VRRP notifications
```

Let's assume that ACME wants the server to receive events related to authentication failure, chassis information (such as power supply health, interface information), operations performed remotely, routing failures, when the device is reloaded, any configuration changes made, and any service failures. The following commands do just that:

```
set snmp trap-group SNMP categories authentication
set snmp trap-group SNMP categories chassis
set snmp trap-group SNMP categories link
set snmp trap-group SNMP categories remote-operations
set snmp trap-group SNMP categories routing
set snmp trap-group SNMP categories startup
set snmp trap-group SNMP categories configuration
set snmp trap-group SNMP categories services
```

Now that the traps have been set, the configuration that allows the SNMP server to send SNMP get requests can be added. The first command in the following configuration sets the device to accept SNMP requests only from the inside interface, which is interface ge-0/0/2.0. The next command sets the community name as NOT-PUBLIC with the permissions being set as read-only. And the final command specifies that Junos OS will only accept requests from a particular subnet or IP address; in this case, access is restricted to the IP address of the SNMP server:

```
set snmp interface ge-0/0/2.0
set snmp community NOT-PUBLIC authorization read-only
set snmp community NOT-PUBLIC clients 10.155.1.40/32
```

Now that the necessary commands have been entered, the configuration can be committed before moving on to test whether everything will work as expected prior to moving the device into the live environment.

Testing the SRX Series Management Configuration

The first thing you need to test is whether devices on the inside can still ping the SRX Series. As the SRX Series is still in the lab, this will be tested from an EX switch that is connected to the inside interface. As you can see, the SRX Series responded without issue:

```
netadmin@ACME-LAB-SW-01> ping 10.100.1.1
PING 10.100.1.1 (10.100.1.1): 56 data bytes
64 bytes from 10.100.1.1: icmp_seq=0 ttl=254 time=3.490 ms
64 bytes from 10.100.1.1: icmp_seq=1 ttl=254 time=3.615 ms
64 bytes from 10.100.1.1: icmp_seq=2 ttl=254 time=4.417 ms
64 bytes from 10.100.1.1: icmp_seq=3 ttl=254 time=3.440 ms
^C
--- 10.100.1.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 3.440/3.740/4.417/0.396 ms
```

Of course, it would be wise to test whether devices outside are unable to ping the SRX Series, which in this case is done from a client on the Internet:

```
NetClient:~ NetUser$ ping 203.0.113.10
PING 203.0.113.10 (203.0.113.10): 56 data bytes
Request timeout for icmp_seq 0
Request timeout for icmp_seq 1
Request timeout for icmp_seq 2
^C
--- 203.0.113.10 ping statistics ---
4 packets transmitted, 0 packets received, 100.0% packet loss
```

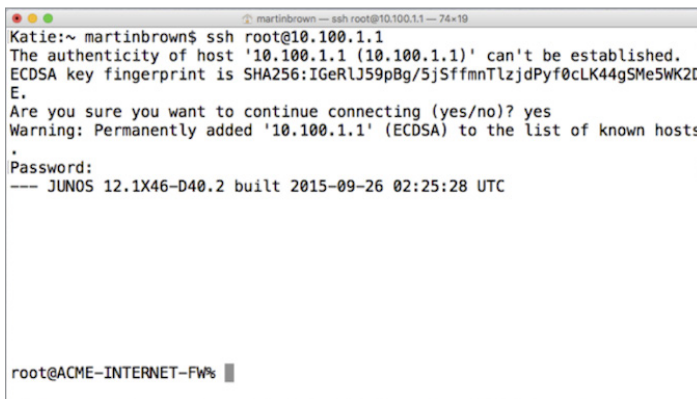
So far so good, the SRX Series rejects the ping and we get a request timeout, so now we can move on and test connectivity. Let's assume that the testing of denying Telnet was successful, and instead an SSH session will be tested from a device with the IP address of 10.0.0.118, as shown in Figure 5.2.



```
martinbrown ~ -bash — 68x17
Katie:~ martinbrown$ ifconfig en4 | grep netmask
    inet 10.0.0.118 netmask 0xfffff00 broadcast 10.0.0.255
Katie:~ martinbrown$ ssh root@10.100.1.1
ssh: connect to host 10.100.1.1 port 22: Connection refused
Katie:~ martinbrown$
```

Figure 5.2 SSH Connection Attempt

As is clearly evident, the connection attempt was refused by the SRX Series, which is what you want to see because the firewall filter specified reject as opposed to discard. Now, let's attempt this from an address that falls within the Management VLAN range, as in Figure 5.3.



```
martinbrown ~ ssh root@10.100.1.1 — 74x10
Katie:~ martinbrown$ ssh root@10.100.1.1
The authenticity of host '10.100.1.1 (10.100.1.1)' can't be established.
ECDSA key fingerprint is SHA256:IGeRLJ59pBg/5jSffmnTlzjdPyf0cLK44gSMe5WK2D
E.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.100.1.1' (ECDSA) to the list of known hosts
Password:
--- JUNOS 12.1X46-D40.2 built 2015-09-26 02:25:28 UTC

root@ACME-INTERNET-FW%
```

Figure 5.3 SSH Connection Allowed

The output shown in Figure 5.3 is interesting. The connection attempt was successful, however the output shows something else. The terminal session asks whether this device can be trusted and shows a *fingerprint*. This fingerprint is useful because if an attacker attempted to spoof a device another error would be displayed stating that the fingerprint has changed, and the connection would immediately be denied.

TIP When the migration from the ASA to the SRX Series is being performed, this fingerprint issue will need to be taken into consideration for any client that has connected to the ASA via SSH.

Most clients will have a record of the fingerprint associated with the IP address 10.100.1.1, which is currently associated with the ASA, and these clients will be connected to the SRX Series, post migration. Therefore, any client that previously performed administration for the ASA will need to remove the fingerprint from their SSH databases before they can successfully connect to the SRX Series.

The SRX Series Can Tell the Time

Checking whether the SRX Series is synchronizing its time with the NTP server is quite easy and can be done with two Junos commands. The first is `show ntp associations`. This tells us what address was configured, what time source that server used, and useful information such as the stratum of that server, any delay, and so on:

```
root@ACME-INTERNET-FW> show ntp associations
remote      refid      st t when poll reach  delay  offset  jitter
=====
*10.255.1.25 .GPS.      1 -  58  64   3   3.235  10.995  1.497
```

In this case, the SRX Series has synced correctly. The command `show ntp status` can also be used to check which NTP server was used, should multiple NTP server statements have been configured:

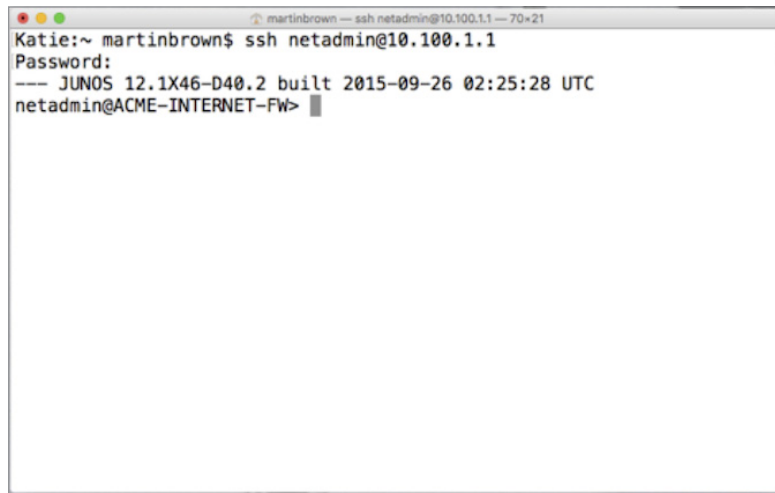
```
root@ACME-INTERNET-FW> show ntp status
status=0664 leap_none, sync_ntp, 6 events, event_peer/strat_chg,
version="ntpd 4.2.0-a Sat Sep 26 04:37:05 UTC 2015 (1)",
processor="octeon", system="JUNOS12.1X46-D40.2", leap=00, stratum=2,
precision=-17, rootdelay=3.235, rootdispersion=3.150, peer=8532,
refid=10.255.1.25,
reftime=db5a20f0.b4409274 Sat, Aug 13 2016 22:42:56.704, poll=6,
clock=db5a212f.640620a7 Sat, Aug 13 2016 22:43:59.390, state=3,
offset=0.000, frequency=0.000, jitter=0.855, stability=0.000
```

Testing AAA Authentication

There is really only one way to test that the TACACS+ configuration has been successful and that is to log in to the device using credentials that are allowed by the ACS server. For testing purposes, the ACS server has been configured with the username *netadmin*, therefore connecting to the SRX Series using these credentials should be successful.

The screen-captured terminal in Figure 5.4 shows a connection attempt via SSH to the IP address 10.100.1.1 from a client within the Management VLAN. The test was successful and you can clearly see the username and the name of the SRX Series.

Once it is confirmed the login was successful, go ahead and attempt to configure the device and try various commands to ensure the both the authentication and the authorization work.

A terminal window titled 'martinbrown — ssh netadmin@10.100.1.1 — 70x21'. The prompt is 'Katie:~ martinbrown\$'. The user enters 'ssh netadmin@10.100.1.1'. The prompt changes to 'Password:'. The user enters a password (indicated by a black box). The terminal displays '--- JUNOS 12.1X46-D40.2 built 2015-09-26 02:25:28 UTC' and the prompt changes to 'netadmin@ACME-INTERNET-FW>'.

```
Katie:~ martinbrown$ ssh netadmin@10.100.1.1
Password:
--- JUNOS 12.1X46-D40.2 built 2015-09-26 02:25:28 UTC
netadmin@ACME-INTERNET-FW>
```

Figure 5.4 Testing AAA Authentication

Is The SRX Series Telling Us It's Alive?

The final configuration steps for the ASA to SRX Series migration were to monitor the SRX Series, and there were two parts to this. The first was to tell the SRX Series to send syslog messages to the syslog server and the second was for the SRX Series to send SNMP traps to the SNMP server, and for the SNMP server to send SNMP gets to the SRX Series.

Checking whether syslog messages are being sent successfully requires the use of some software that is able to receive syslog messages, and for the SRX Series to have something to send to the syslog server. With the Junos OS, the syslog messages sent when a commit is performed are quite verbose, as you can see in Figure 5.5. Therefore, there is no need to wait for any events to occur, you just need to perform a commit.

Figure 5.5 shows an example of syslog messages received by the Kiwi Syslog Server, running on a Windows server. Although these messages did not originate from our SRX Series, they demonstrate the types of messages that are sent during a commit within the Junos OS.

Date	Time	Priority	Hostname	Message
01-21-2015	10:54:48	Conn Info	192.168.0.1	Jan 21 10:54:48 SRC conn[1402]: [OID] (/usr/libexec/atscan)
01-21-2015	10:54:47	Conn Info	192.168.0.254	Jan 21 10:55:00 SW conn[1402]: [OID] (/usr/libexec/atscan)
01-21-2015	10:53:17	Local Info	192.168.0.254	Jan 21 10:53:30 SW SWITCH exp[1232]: UO_COMMIT_PROGRESS: Commit operation in progress: signaling 'Alarm control process', and
01-21-2015	10:53:17	Local Warning	192.168.0.254	Jan 21 10:53:30 SW SWITCH exp[1232]: UO_COMMIT_COMPLETED: commit complete
01-21-2015	10:53:17	Local Info	192.168.0.254	Jan 21 10:53:30 SW SWITCH exp[1232]: UO_COMMIT_PROGRESS: Commit operation in progress: commit complete
01-21-2015	10:53:17	Local Info	192.168.0.254	Jan 21 10:53:30 SW SWITCH exp[1232]: UO_COMMIT_PROGRESS: Commit operation in progress: child notifying daemon of new config
01-21-2015	10:53:17	Local Info	192.168.0.254	Jan 21 10:53:30 SW SWITCH exp[1232]: UO_COMMIT_PROGRESS: Commit operation in progress: notifying daemon of new configuration
01-21-2015	10:53:17	Local Info	192.168.0.254	Jan 21 10:53:30 SW SWITCH exp[1232]: UO_COMMIT_PROGRESS: Commit operation in progress: updating 'Simple Network Management
01-21-2015	10:53:17	Local Info	192.168.0.254	with notification owner enabled
01-21-2015	10:53:17	Local Info	192.168.0.254	Jan 21 10:53:30 SW SWITCH exp[1232]: UO_COMMIT_PROGRESS: Commit operation in progress: fileback hooking
01-21-2015	10:53:17	Local Info	192.168.0.254	Jan 21 10:53:30 SW SWITCH exp[1232]: UO_COMMIT_PROGRESS: Commit operation in progress: activating '/usr/libexec/unique data'
01-21-2015	10:53:17	Local Info	192.168.0.254	Commit operation in progress: db_group_info_clear done
01-21-2015	10:53:17	Local Info	192.168.0.254	Jan 21 10:53:30 SW SWITCH exp[1232]: UO_COMMIT_PROGRESS: Commit operation in progress: db_group_info_clear start
01-21-2015	10:53:17	Local Info	192.168.0.254	Jan 21 10:53:30 SW SWITCH exp[1232]: UO_COMMIT_PROGRESS: Commit operation in progress: db_check_constant_info_clear done
01-21-2015	10:53:17	Local Info	192.168.0.254	Commit operation in progress: db_check_constant_info_clear start
01-21-2015	10:53:17	Local Info	192.168.0.254	Jan 21 10:53:30 SW SWITCH exp[1232]: UO_COMMIT_PROGRESS: Commit operation in progress: copying configuration to jumper.care
01-21-2015	10:53:17	Local Info	192.168.0.254	Jan 21 10:53:30 SW SWITCH exp[1232]: UO_COMMIT_PROGRESS: Commit operation in progress: finish file activate
01-21-2015	10:53:16	Local Info	192.168.0.254	Commit operation in progress: not connecting to conn in vti
01-21-2015	10:53:16	Local Info	192.168.0.254	Jan 21 10:53:29 SW SWITCH exp[1232]: UO_COMMIT_PROGRESS: Commit operation in progress: executing function, commands
01-21-2015	10:53:16	Local Info	192.168.0.254	Jan 21 10:53:29 SW SWITCH exp[1232]: UO_COMMIT_PROGRESS: Commit operation in progress: activating '/usr/libexec/cont'
01-21-2015	10:53:16	Local Info	192.168.0.254	Jan 21 10:53:29 SW SWITCH exp[1232]: UO_COMMIT_PROGRESS: Commit operation in progress: activating '/usr/libexec/cont'
01-21-2015	10:53:16	Local Info	192.168.0.254	Jan 21 10:53:29 SW SWITCH exp[1232]: UO_COMMIT_PROGRESS: Commit operation in progress: activating '/usr/libexec/pan.conf'
01-21-2015	10:53:16	Local Info	192.168.0.254	Jan 21 10:53:29 SW SWITCH exp[1232]: UO_COMMIT_PROGRESS: Cleanup child 'usr/libexec/ly, PD 1246, status 0
01-21-2015	10:53:15	Local Info	192.168.0.254	Jan 21 10:53:28 SW SWITCH exp[1232]: UO_COMMIT_STATUS: Stating child 'usr/libexec/ly'
01-21-2015	10:53:15	Local Info	192.168.0.254	Jan 21 10:53:28 SW SWITCH exp[1232]: UO_COMMIT_PROGRESS: Commit operation in progress: start file activate
01-21-2015	10:53:15	Local Info	192.168.0.254	Jan 21 10:53:28 SW SWITCH exp[1232]: UO_COMMIT_PROGRESS: Commit operation in progress: commit success
01-21-2015	10:53:15	Local Info	192.168.0.254	Jan 21 10:53:28 SW SWITCH exp[1232]: UO_COMMIT_STATUS: Cleanup child 'usr/libexec/ly, PD 1246, status 0
01-21-2015	10:53:14	Local Info	192.168.0.254	Jan 21 10:53:27 SW SWITCH exp[1232]: UO_COMMIT_STATUS: Stating child 'usr/libexec/ly'
01-21-2015	10:53:14	Local Info	192.168.0.254	Commit operation in progress: daemon checking new configuration
01-21-2015	10:53:14	Local Info	192.168.0.254	Jan 21 10:53:27 SW SWITCH exp[1232]: UO_COMMIT_STATUS: Cleanup child 'usr/libexec/ly, PD 1236, status 0
01-21-2015	10:53:14	Local Info	192.168.0.254	Jan 21 10:53:26 SW SWITCH exp[1232]: UO_COMMIT_STATUS: Stating child 'usr/libexec/ly'
01-21-2015	10:53:13	Local Info	192.168.0.254	Jan 21 10:53:26 SW SWITCH exp[1232]: UO_COMMIT_PROGRESS: Commit operation in progress: dropping unchanged foreign links
01-21-2015	10:53:13	Local Info	192.168.0.254	Jan 21 10:53:26 SW SWITCH exp[1232]: UO_COMMIT_PROGRESS: Commit operation in progress: complete function file

Figure 5.5 Syslog Messages Following a Commit

On the other hand, SNMP is slightly different in that there is a command within the Junos OS that allows an administrator to see a summary of the SNMP traps that have been sent by Junos OS, and also a count of the number of get requests the SRX Series has received. The command is `show snmp statistics` and your output would be similar to the following:

```

netadmin@ACME-INTERNET-FW> show snmp statistics
SNMP statistics:
Input:
Packets: 1885, Bad versions: 0, Bad community names: 0,
Bad community uses: 0, ASN parse errors: 0,
Too bigs: 0, No such names: 0, Bad values: 0,
Read onlys: 0, General errors: 0,
Total request varbinds: 6479, Total set varbinds: 0,
Get requests: 1550, Get nexts: 335, Set requests: 0,
Get responses: 0, Traps: 0,
Silent drops: 0, Proxy drops: 0, Commit pending drops: 0,
Throttle drops: 0, Duplicate request drops: 0
V3 Input:
Unknown security models: 0, Invalid messages: 0
Unknown pdu handlers: 0, Unavailable contexts: 0
Unknown contexts: 0, Unsupported security levels: 0
Not in time windows: 0, Unknown user names: 0
Chapter 2: SNMP and the Health of Your Network 27
Unknown engine ids: 0, Wrong digests: 0, Decryption errors: 0
Output:
Packets: 1897, Too bigs: 0, No such names: 0,
Bad values: 0, General errors: 0,
Get requests: 0, Get nexts: 0, Set requests: 0,
Get responses: 1885, Traps: 12

```

Conclusion: The Migration Process

By now you are ready to bring the SRX Series into service. The tests you performed were successful and demonstrated that the device should perform as expected. “What else do I need to do?” you might ask.

First, you need to document ACME’s QA process (let’s assume one already exists). This process should include recommended software versions and settings, and it should ensure that important security features, such as a firewall filter applied to lo0.0, have been configured.

Second, you need to consider the actual change process, which includes a detailed plan of the steps you need to take during your migration from the ASA to the SRX Series:

*Unplug the Ethernet cable labeled R.123 from g0/0 on ASA
ACME-INTERNET-FW.*

*Plug the Ethernet cable labeled R.123 into ge-0/0/0.0 on SRX
ACME-INTERNET-FW.*

*Unplug the Ethernet cable labeled R.124 from g0/1 on ASA
ACME-INTERNET-FW.*

*Plug the Ethernet cable labeled R.124 into ge-0/0/1.0 on SRX
ACME-INTERNET-FW.*

*Unplug the Ethernet cable labeled R.125 from g0/2 on ASA
ACME-INTERNET-FW.*

*Plug the Ethernet cable labeled R.125 into ge-0/0/2.0 on SRX
ACME-INTERNET-FW.*

Connect to the SRX Series ACME-INTERNET-FW via SSH.

Check to ensure the VPN connection to branch office is up and devices in the branch are reachable.

....

While this plan may appear simple, it is still important to list every single step – including the necessary tests that need to be performed and a rollback plan – so that every engineer involved with the process knows what needs to be done next, how they can confirm everything is working as expected, and, more importantly, what to do to prevent the migration from failing.

With a detailed change process ACME can replace all the ASAs and upgrade their perimeter to the SRX Series, at any time, by using any staff engineer. It’s day one and you have a job to do.

Book Two: Migrating From Cisco Catalyst to Juniper EX Series

By Martin Brown & Rob Jeffery

Chapter 6: IOS to Junos Configuration 64

Chapter 7: VLANs and RVIs 72

Chapter 8: Link Aggregation.....80

Chapter 9: Access to Trunk Ports.....85

Chapter 10: Redundancy 95

Chapter 11: Security Hardening101

Conclusion..... 105

What You Need to Know Before Reading This Book

Before reading this book, you should be familiar with the basic administrative functions of the Junos operating system, including the ability to work with operational commands and to read, understand, and change Junos configurations. There are several books in the *Day One* library on learning Junos, at www.juniper.net/dayone.

Audience

This book is intended for network engineers who have just begun their career in network engineering, and while they are aware of the various routing protocols, they are perhaps unsure of the features each one has to offer.

We have also written this book for network engineers who have a year's experience in supporting live networks but have only had exposure to maybe one or two routing protocols. This book will help engineers learn more about most of the IPv4 routing protocols Junos OS has to offer.

What You Need to Know Before Reading This Book

Before reading this book, you should be familiar with the basic administrative functions of the Junos operating system, including the ability to work with operational commands and to read, understand, and change Junos configurations. There are several books in the *Day One* library on learning Junos, at www.juniper.net/dayone.

By Reading This Book You Will

- Better understand the different interior gateway protocols
- Translate IOS commands to Junos OS commands
- Learn the differences between HSRP and VRRP for gateway redundancy
- Understand how a Virtual Chassis can ease network management

Information Experience

This *Day One* book is singularly focused on one aspect of networking technology that you might be able to deploy in one day. There are many other sources of information at Juniper Networks, from white papers, to webinars, to online forums such as J-Net (forums.juniper.net). Look for the following MORE? sidebars for direct access to other superb resources for information.

MORE? It's highly recommended you go through the technical documentation in order to become fully acquainted with the initial configuration process of Junos devices in order to get a better understanding of the configuration process flow, before you jump in. The technical documentation is located at: <http://www.juniper.net/documentation>. Use the Pathfinder tool on the documentation site to explore and find the right information for your needs.

MORE? For more details and tutorials on the EX Series, see: *Day One: EX Series Up and Running* at <https://www.juniper.net/us/en/training/jnbooks/day-one/fabric-switching-tech-series/ex-series-up-running/>.

Preface

There are tens of thousands of medium- to-large companies around the globe, and it is almost guaranteed that each of these is running some kind of switched network. In fact, you would have to think very hard to imagine even a small company without any form of network—maybe a monastery that brews beer—but beyond something like that, it seems most companies need at least a switched network.

So, whatever the size or intent of those tens of thousands of companies, their switched networks need to be replaced periodically to avoid failure, or upgraded to take advantage of new technologies, or faster speeds, or lower power consumption. During the planning stage of this migration, a decision needs to be made as to whether to replace the network devices with ones from the same manufacturer or move to another manufacturer that offers faster speeds, or whose switches adhere more to recognized standards.

This book imagines a scenario in which a medium-sized company, called ACME, has decided to upgrade their access layer switches. Their current switches have been reliable over the years, but after running a comparison, they have decided to move from a Cisco Catalyst switch to a Juniper EX Series switch.

At the moment, ACME is only planning on replacing the access layer, as the distribution and core layers are reasonably new. Therefore the migration must ensure that come the day of the cutover from the old to the new devices, everything will work as before without any end users noticing any changes apart from a possible increase in network speed.

To begin with, the new EX Series switches are being built without any of the interfaces being connected to the existing network. This is because the IP addresses and VLAN numbers in use must match those configured on the existing network and by placing these devices on the live network, address conflicts will occur. This doesn't pose a huge issue, however, as the devices are being configured using a console cable and the switches will go through thorough testing before the cutover date.

Figure P.1 diagrams what ACME's network topology currently looks like. The switch we are replacing is "ACME-Cisco-SW-10" which is in the access layer. This is connected to a second access layer switch, "ACME-Cisco-SW-11", via an aggregated link or port channel. Switch SW-11 is currently being built by a colleague, so we are only interested in building SW-10. Switches SW-10 and SW-11 are in turn connected to the aggregation layer switch via another two-port channel.

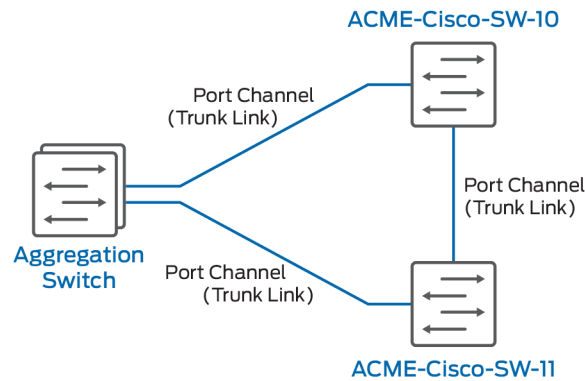


Figure P.1 ACME's Network Topology

The ports channels that are being used to link the switches together are configured as trunk links. This means these links carry the traffic from multiple VLANs. There are in fact six VLANs configured on the access layer switches:

- Sales
- Finance
- Engineering
- Management
- Transit
- Unused

The Sales, Finance and Engineering, VLANs allow workstations to connect to the network. The Management VLAN allows network engineers to monitor and configure the network devices. The Transit VLAN carries traffic from the access layer to the aggregation layer, which in turn forwards it towards the core.

The Unused VLAN, however, is unlike the others in that it doesn't pass any traffic. It is a security mechanism and any interfaces that are spare are added to this VLAN so that if an intruder attaches a device to a spare interface, there is no possibility the intruder can access any network resources.

The network devices in the access layer are 24-port Cisco Catalyst 3560X switches configured as a Layer 3 switch so that they are also performing routing as well as switching duties. The plan is to replace these switches with Juniper EX Series switches.

NOTE The IOS version running on the Cisco Catalyst switch is 15.2(4)E4. The version of Junos OS the EX Series is using is in-line with the JTAC version recommended during the time this book was being written: version 15.1R5.5. However, most of the commands used in this book will be version neutral, applicable to almost any version of Junos.

As you progress through this book, the ACME environment will become much more apparent, so by the time you have fully converted the configuration from the Catalyst to the new EX Series switch, you should be completely familiar with ACME's environment. For now, however, relax, kick back, and prepare to learn all about converting IOS VLANs and SVIs into Junos OS VLANs and RVIs. Enjoy the migration.

Martin Brown and Rob Jeffery

Chapter 6

IOS to Junos Configuration

Catalyst to EX Series Configuration

```
IOS
Current configuration : 8818 bytes
!
!
version 15.2
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
!
hostname ACME-Cisco-SW-10
!
boot-start-marker
boot-end-marker
!
!
enable secret 5 $1$wmPG$H2yhSxIbKSxdyaRsD5oFZ0
!
username admin privilege 15 password 7
03345A1815182E5E4A58
aaa new-model
!
!
aaa authentication login default local
aaa authentication enable default enable
!
!
!
!
!
aaa session-id common
```

```
JUNOS
version 15.1R5.5;
system {
    host-name ACME-Juniper-SW-10;
    domain-name ACME.com;
    root-authentication {
        encrypted-password "$5$/oGYh9yi$ "; ##
SECRET-DATA
    }
    login {
        message "*****\n*
*****\n*
*\n* This is a private network device. If you *
*\n* are not authorised to connect to this
*\n* device, please disconnect immediately.
*\n*
*\n*
*****";
        user admin {
            uid 2001;
            class super-user;
            authentication {
                encrypted-password.4nNR13oZI."; ##
SECRET-DATA
            }
        }
    }
    services {
        ssh {
            protocol-version v2;
        }
    }
    ntp {
```

```

system mtu routing 1500
!
!
!
!
ip routing
!
!
!
ip domain-name ACME.com
vtp mode transparent
!
!
!
!
!
!
license boot level ipservices
!
!
!
!
spanning-tree mode rapid-pvst
spanning-tree extend system-id
spanning-tree vlan 1-4094 priority 8192
!
!
!
!
vlan internal allocation policy ascending
!
vlan 10
  name FINANCE
!
vlan 20
  name SALES
!
vlan 30
  name ENGINEERING
!
vlan 100
  name MANAGEMENT
!
vlan 200
  name TRANSIT
!
vlan 900
  name UNUSED
!
!
!
!
!
!
!
!
!

```

```

    server 10.200.0.123 prefer;
  }
}
chassis {
  aggregated-devices {
    ethernet {
      device-count 2;
    }
  }
}
interfaces {
  interface-range UNUSED {
    member-range ge-0/0/9 to ge-0/0/23;
    disable;
    unit 0 {
      description <<UNUSED>>;
      family ethernet-switching {
        vlan {
          members UNUSED;
        }
      }
    }
  }
}
ge-0/0/0 {
  unit 0 {
    description <<ESXI-SERVER-01>>;
    family ethernet-switching {
      port-mode trunk;
      vlan {
        members [ 10 20 30 ];
      }
      native-vlan-id 900;
    }
  }
}
ge-0/0/1 {
  unit 0 {
    description <<ESXI-SERVER-02>>;
    family ethernet-switching {
      port-mode trunk;
      vlan {
        members [ 10 20 30 ];
      }
      native-vlan-id 900;
    }
  }
}
ge-0/0/2 {
  unit 0 {
    description <<WORKSTATION1>>;
    family ethernet-switching {
      vlan {
        members 10;
      }
    }
  }
}
ge-0/0/3 {
  unit 0 {

```

```

!
!
!
!
interface Port-channel1
description <<UPLINK_TO_AGG_LAYER>>
switchport trunk allowed vlan 100,200
switchport trunk encapsulation dot1q
switchport trunk native vlan 900
switchport mode trunk
!
interface Port-channel2
description <<UPLINK_TO_SW-11>>
switchport trunk allowed vlan 10,20,30,100,200
switchport trunk encapsulation dot1q
switchport trunk native vlan 900
switchport mode trunk
!
interface FastEthernet0
no ip address
no ip route-cache
shutdown
!
interface GigabitEthernet0/1
description <<ESXI-SERVER-01>>
switchport trunk allowed vlan 10,20,30
switchport trunk encapsulation dot1q
switchport trunk native vlan 900
switchport mode trunk
!
interface GigabitEthernet0/2
description <<ESXI-SERVER-02>>
switchport trunk allowed vlan 10,20,30
switchport trunk encapsulation dot1q
switchport trunk native vlan 900
switchport mode trunk
!
interface GigabitEthernet0/3
description <<WORKSTATION1>>
switchport access vlan 10
switchport port-security maximum 2
switchport port-security violation restrict
switchport port-security mac-address sticky
!
interface GigabitEthernet0/4
description <<WORKSTATION2>>
switchport access vlan 10
switchport port-security maximum 2
switchport port-security violation restrict
switchport port-security mac-address sticky
!
interface GigabitEthernet0/5
description <<WORKSTATION3>>
switchport access vlan 10
switchport port-security maximum 2
switchport port-security violation restrict
switchport port-security mac-address sticky
!

```

```

description <<WORKSTATION2>>;
family ethernet-switching {
vlan {
members 10;
}
}
}
ge-0/0/4 {
unit 0 {
description <<WORKSTATION3>>;
family ethernet-switching {
vlan {
members 10;
}
}
}
}
ge-0/0/5 {
unit 0 {
description <<WORKSTATION4>>;
family ethernet-switching {
vlan {
members 20;
}
}
}
}
ge-0/0/6 {
unit 0 {
description <<WORKSTATION5>>;
family ethernet-switching {
vlan {
members 20;
}
}
}
}
ge-0/0/7 {
unit 0 {
description <<WORKSTATION6>>;
family ethernet-switching {
vlan {
members 30;
}
}
}
}
ge-0/0/8 {
unit 0 {
description <<WORKSTATION7>>;
family ethernet-switching {
vlan {
members 30;
}
}
}
}
}

```

67 Catalyst to EX Series Configuration

```
interface GigabitEthernet0/6
description <<WORKSTATION4>>
switchport access vlan 20
switchport port-security maximum 2
switchport port-security violation restrict
switchport port-security mac-address sticky
!
interface GigabitEthernet0/7
description <<WORKSTATION5>>
switchport access vlan 20
switchport port-security maximum 2
switchport port-security violation restrict
switchport port-security mac-address sticky
!
interface GigabitEthernet0/8
description <<WORKSTATION6>>
switchport access vlan 30
switchport port-security maximum 2
switchport port-security violation restrict
switchport port-security mac-address sticky
!
interface GigabitEthernet0/9
description <<WORKSTATION7>>
switchport access vlan 30
switchport port-security maximum 2
switchport port-security violation restrict
switchport port-security mac-address sticky
!
interface GigabitEthernet0/10
description <<UNUSED>>
switchport access vlan 900
shutdown
!
interface GigabitEthernet0/11
description <<UNUSED>>
switchport access vlan 900
shutdown
!
interface GigabitEthernet0/12
description <<UNUSED>>
switchport access vlan 900
shutdown
!
interface GigabitEthernet0/13
description <<UNUSED>>
switchport access vlan 900
shutdown
!
interface GigabitEthernet0/14
description <<UNUSED>>
switchport access vlan 900
shutdown
!
interface GigabitEthernet0/15
description <<UNUSED>>
switchport access vlan 900
shutdown
!
```

```
ge-0/0/9 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/10 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/11 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/12 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/13 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/14 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/15 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/16 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/17 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/18 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/19 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/20 {
    unit 0 {
        family ethernet-switching;
    }
}
```

```
interface GigabitEthernet0/16
description <<UNUSED>>
switchport access vlan 900
shutdown
!
interface GigabitEthernet0/17
description <<UNUSED>>
switchport access vlan 900
shutdown
!
interface GigabitEthernet0/18
description <<UNUSED>>
switchport access vlan 900
shutdown
!
interface GigabitEthernet0/19
description <<UNUSED>>
switchport access vlan 900
shutdown
!
interface GigabitEthernet0/20
description <<UNUSED>>
switchport access vlan 900
shutdown
!
interface GigabitEthernet0/21
description <<UNUSED>>
switchport access vlan 900
shutdown
!
interface GigabitEthernet0/22
description <<UNUSED>>
switchport access vlan 900
shutdown
!
interface GigabitEthernet0/23
description <<UNUSED>>
switchport access vlan 900
shutdown
!
interface GigabitEthernet0/24
description <<UNUSED>>
switchport access vlan 900
shutdown
!
interface GigabitEthernet1/1
description <<PORT CHANNEL 1>>
switchport trunk allowed vlan 100,200
switchport trunk encapsulation dot1q
switchport trunk native vlan 900
switchport mode trunk
channel-group 1 mode active
!
interface GigabitEthernet1/2
description <<PORT CHANNEL 1>>
switchport trunk allowed vlan 100,200
switchport trunk encapsulation dot1q
switchport trunk native vlan 900
```

```

}
}
ge-0/0/21 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/22 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/23 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/1/0 {
    description <<AGGREGATED-ETHERNET-0>>;
    ether-options {
        802.3ad ae0;
    }
}
ge-0/1/1 {
    description <<AGGREGATED-ETHERNET-0>>;
    ether-options {
        802.3ad ae0;
    }
}
ge-0/1/2 {
    description <<AGGREGATED-ETHERNET-1>>;
    ether-options {
        802.3ad ae1;
    }
}
ge-0/1/3 {
    description <<AGGREGATED-ETHERNET-1>>;
    ether-options {
        802.3ad ae1;
    }
}
ae0 {
    unit 0 {
        description <<UPLINK_TO_AGG_LAYER>>;
        family ethernet-switching {
            port-mode trunk;
            vlan {
                members [ 100 200 ];
            }
            native-vlan-id 900;
        }
    }
}
ae1 {
    unit 0 {
        description <<UPLINK_TO_SW-11>>;
        family ethernet-switching {
            port-mode trunk;

```

69 Catalyst to EX Series Configuration

```

switchport mode trunk
channel-group 1 mode active
!
interface GigabitEthernet1/3
description <<PORT CHANNEL 2>>
switchport trunk allowed vlan 10,20,30,100,200
switchport trunk encapsulation dot1q
switchport trunk native vlan 900
switchport mode trunk
channel-group 2 mode active
!
interface GigabitEthernet1/4
description <<PORT CHANNEL 2>>
switchport trunk allowed vlan 10,20,30,100,200
switchport trunk encapsulation dot1q
switchport trunk native vlan 900
switchport mode trunk
channel-group 2 mode active
!
interface TenGigabitEthernet1/1
!
interface TenGigabitEthernet1/2
!
interface Vlan1
no ip address
shutdown
!
interface Vlan10
ip address 192.168.10.10 255.255.255.0
standby version 2
standby 10 ip 192.168.10.1
standby 10 priority 110
standby 10 preempt
!
interface Vlan20
ip address 192.168.20.10 255.255.255.0
standby version 2
standby 20 ip 192.168.20.1
standby 20 priority 110
standby 20 preempt
!
interface Vlan30
ip address 192.168.30.10 255.255.255.0
standby version 2
standby 30 ip 192.168.30.1
standby 30 priority 110
standby 30 preempt
!
interface Vlan100
ip address 10.100.0.10 255.255.255.0
!
interface Vlan200
ip address 192.168.200.10 255.255.255.0
standby version 2
standby 200 ip 192.168.200.12
standby 200 priority 110
standby 200 preempt
!

```

```

        vlan {
            members [ 10 20 30 100 200 ];
        }
        native-vlan-id 900;
    }
}
lo0 {
    unit 0 {
        family inet {
            filter {
                input MAN_FILTER;
            }
        }
    }
}
me0 {
    unit 0 {
        family inet;
    }
}
vlan {
    unit 10 {
        family inet {
            address 192.168.10.10/24 {
                vrrp-group 10 {
                    virtual-address 192.168.10.1;
                    priority 110;
                    preempt;
                }
            }
        }
    }
    unit 20 {
        family inet {
            address 192.168.20.10/24 {
                vrrp-group 20 {
                    virtual-address 192.168.20.1;
                    priority 110;
                    preempt;
                }
            }
        }
    }
    unit 30 {
        family inet {
            address 192.168.30.10/24 {
                vrrp-group 30 {
                    virtual-address 192.168.30.1;
                    priority 110;
                    preempt;
                }
            }
        }
    }
    unit 100 {
        family inet {
            address 10.100.0.10/24;
        }
    }
}

```

```
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
ip route 0.0.0.0 0.0.0.0 192.168.200.1
ip route 10.200.0.0 255.255.255.0 10.100.0.1
!
ip access-list extended MAN_ACL
permit ip 10.200.0.0 0.0.0.255 any
!
!
!
!
!
!
banner login ^C
*****
*                               *
* This is a private network device. If you *
* are not authorised to connect to this *
* device, please disconnect immediately. *
*                               *
*****
^C
!
line con 0
line vty 0 4
access-class MAN_ACL in
transport input ssh
line vty 5 15
access-class MAN_ACL in
transport input ssh
!
ntp server 10.200.0.123 prefer
!
end
```

```

    }
  }
  unit 200 {
    family inet {
      address 192.168.200.10/24 {
        vrrp-group 200 {
          virtual-address
192.168.200.12;
          priority 110;
          preempt;
        }
      }
    }
  }
}
routing-options {
  static {
    route 0.0.0.0/0 next-hop 192.168.200.1;
    route 10.200.0.0/24 next-hop 10.100.0.1;
  }
}
protocols {
  igmp-snooping {
    vlan all;
  }
  rstp {
    bridge-priority 8k;
  }
  lldp {
    interface all;
  }
  lldp-med {
    interface all;
  }
}
firewall {
  family inet {
    filter MAN_FILTER {
      term 1 {
        from {
          source-address {
            10.200.0.0/24;
          }
        }
        then accept;
      }
    }
  }
}
ethernet-switching-options {
  secure-access-port {
    interface ge-0/0/2.0 {
      mac-limit 2;
      persistent-learning;
    }
    interface ge-0/0/3.0 {
      mac-limit 2;
    }
  }
}
```

```

        persistent-learning;
    }
    interface ge-0/0/4.0 {
        mac-limit 2;
        persistent-learning;
    }
    interface ge-0/0/5.0 {
        mac-limit 2;
        persistent-learning;
    }
    interface ge-0/0/6.0 {
        mac-limit 2;
        persistent-learning;
    }
    interface ge-0/0/7.0 {
        mac-limit 2;
        persistent-learning;
    }
    interface ge-0/0/8.0 {
        mac-limit 2;
        persistent-learning;
    }
    }
    storm-control {
        interface all;
    }
}
vlans {
    ENGINEERING {
        vlan-id 30;
        l3-interface vlan.30;
    }
    FINANCE {
        vlan-id 10;
        l3-interface vlan.10;
    }
    MANAGEMENT {
        vlan-id 100;
        l3-interface vlan.100;
    }
    SALES {
        vlan-id 20;
        l3-interface vlan.20;
    }
    TRANSIT {
        vlan-id 200;
        l3-interface vlan.200;
    }
    UNUSED {
        vlan-id 900;
    }
}

```

Chapter 7

VLANs and RVIs

Typically, when configuring a network device such as a switch, there is a logical order to be followed in order to complete the configuration efficiently and successfully. For example, you cannot link switches together until you've created trunk ports, and you cannot configure which VLANs will traverse that link until the VLANs themselves have been created.

In this chapter, you will therefore create the VLANs that exist on the existing Cisco Catalyst switch on the Juniper EX Series switch, after which the Layer 3 interfaces will be created and linked to the VLAN. Figure 7.1 is a logical representation of the VLANs that ACME is using in their existing environment and that you need to mirror in the new environment.

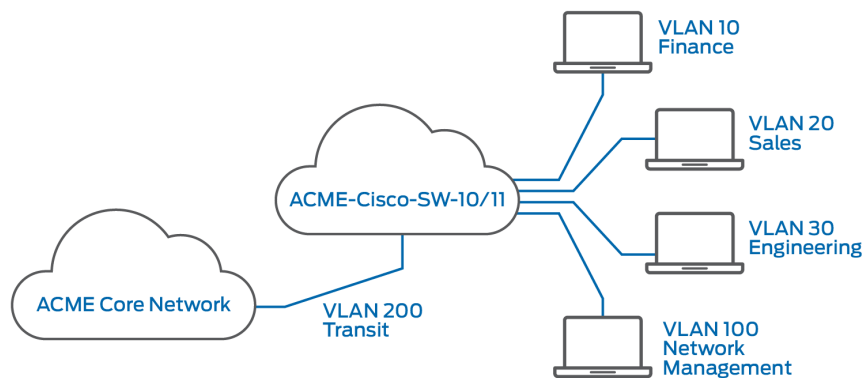


Figure 7.1

Layer 2 Logical Topology

Removing the Default VLAN

When an EX Series switch is first powered on, Junos creates a default VLAN and a corresponding Layer 3 interface, *vlan.0*. Cisco Catalyst switches also create a VLAN, known as VLAN 1 by default, and a Layer 3 interface with the same name.

It is a recommended security best practice not to use the default VLANs on either device. The difference between IOS and Junos OS-based switches, in this respect, is that in IOS you can only disable VLAN 1, whereas within the Junos OS, you can delete the default VLAN. Since it's possible to do that, let's make that the first step in configuring VLANs. The following commands will perform just that task:

```
delete interface vlan.0
delete vlan default
```

Creating the Layer 2 VLANs

Once done, you can begin creating the Layer 2 VLANs. In IOS, the VLAN is given a number and then a name. In our case, the switch is configured with six VLANs: 10, 20, 30, 100, 200, and 900:

```
vlan 10
  name FINANCE
!
vlan 20
  name SALES
!
vlan 30
  name ENGINEERING
!
vlan 100
  name MANAGEMENT
!
vlan 200
  name TRANSIT
!
vlan 900
  name UNUSED
```

Unused interfaces are made part of VLAN 900 as a security best practice, and as such, this VLAN has the name of UNUSED. VLANs 10, 20, and 30 are locally significant and are given the names, FINANCE, SALES, and ENGINEERING, respectively. VLAN 100 is used for managing network devices and is therefore called MANAGEMENT, and VLAN 200 is used as a transit VLAN to the aggregation layer, and as such has the name TRANSIT. VLANs in Junos OS are created using a name. If traffic from multiple VLANs is to traverse a trunk link, then these VLANs must be given a corresponding number called a VLAN ID. This ID is assigned using the *vlan-id* command option:

```
set vlans FINANCE vlan-id 10
set vlans SALES vlan-id 20
```

```
set vlans ENGINEERING vlan-id 30
set vlans MANAGEMENT vlan-id 100
set vlans TRANSIT vlan-id 200
set vlans UNUSED vlan-id 900
```

Creating the Layer 3 Interfaces

Now that the Layer 2 VLANs have been created, attention can turn to the Layer 3 interfaces. Cisco refers to these interfaces as SVIs. In Junos OS, these interfaces are referred to as *Router VLAN Interfaces* (RVIs). Figure 7.2 illustrates which subnets are allocated to which VLAN and SVI.

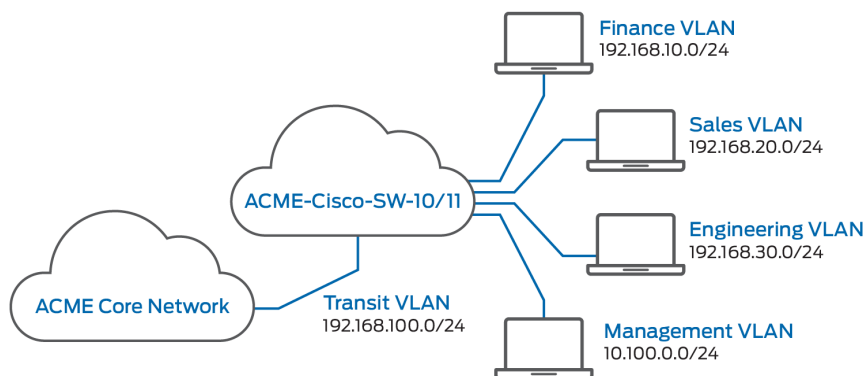


Figure 7.2 Subnet Allocation

In both IOS and Junos, SVIs and RVIs are created under the `interface` hierarchy. IOS matches the interface number to the corresponding Layer 2 VLAN number automatically:

```
interface Vlan10
  ip address 192.168.10.10 255.255.255.0
!
interface Vlan20
  ip address 192.168.20.10 255.255.255.0
!
interface Vlan30
  ip address 192.168.30.10 255.255.255.0
!
interface Vlan100
  ip address 10.100.0.10 255.255.255.0
!
interface Vlan200
  ip address 192.168.200.10 255.255.255.0
```

TIP VLAN 900 was created as part of a security mechanism. This means that there is no need to create a corresponding SVI for this VLAN.

In Junos OS, VLANs are created by name, therefore when an RVI is created, the first step is to link the Layer 2 VLAN to the RVI that the VLAN is to use. This link is made using the option `l3-interface` under the `vlan` hierarchy:

```
set vlans FINANCE l3-interface vlan.10
set vlans SALES l3-interface vlan.20
set vlans ENGINEERING l3-interface vlan.30
set vlans MANAGEMENT l3-interface vlan.100
set vlans TRANSIT l3-interface vlan.200
```

Now that Junos knows which RVI is linked to which Layer 2 interface, you can go ahead and assign the IP addresses to the RVIs. In IOS, the SVIs are created as individual interfaces, whereas in Junos, the RVIs are created as a logical interface under the single VLAN interface.

In Junos OS, subnet masks are configured as a prefix. For example, whereas in IOS a 24-bit subnet mask would be written `255.255.255.0`, in Junos, the same would be written `/24`. The commands to assign IP addresses to the RVIs are as follows:

```
set interfaces vlan unit 10 family inet address 192.168.10.10/24
set interfaces vlan unit 20 family inet address 192.168.20.10/24
set interfaces vlan unit 30 family inet address 192.168.30.10/24
set interfaces vlan unit 100 family inet address 10.100.0.10/24
set interfaces vlan unit 200 family inet address 192.168.200.10/24
```

All that remains now is to commit the changes and if the commit is successful, check that the VLANs have been created. This can be done by using the `show vlans brief` Junos command. This command not only shows the names of the VLAN but the VLAN ID assigned to that number:

```
{master:0}
admin@ACME-Juniper-SW-10> show vlans brief
```

Name	Tag	Primary Address	Ports Active/Total
ENGINEERING	30	192.168.30.10/24	0/0
FINANCE	10	192.168.10.10/24	0/0
MANAGEMENT	100	10.100.0.10/24	0/0
SALES	20	192.168.20.10/24	0/0
TRANSIT	200	192.168.200.10/24	0/0
UNUSED	900		0/0
default			

Once you have confirmed that the VLANs are correctly configured, the `show interfaces terse` command lists all of the interfaces that have been configured within Junos OS and the assigned IP addresses. If the option `| match vlan` is added to the end of this command, the output will only show the RVIs and will filter out the physical interfaces:

```
{master:0}
admin@ACME-Juniper-SW-10> show interfaces terse | match vlan
```

vlan	up	up	
vlan.10	up	down	inet 192.168.10.10/24
vlan.20	up	down	inet 192.168.20.10/24
vlan.30	up	down	inet 192.168.30.10/24
vlan.100	up	down	inet 10.100.0.10/24
vlan.200	up	down	inet 192.168.200.10/24

You may notice that the RVIs are currently showing as *down*. This is because no interfaces have been assigned to those VLANs, and as such, they are unable to come up just yet. With the exception of the interfaces being down, you can still see that the VLANs and RVIs have been configured correctly, meaning you can now move on to the next task which involves telling Junos OS how to reach other parts of the LAN.

Configuring Static Routes

RVIs only tell the network device which subnets are locally attached. They do not, however, tell the switch how to connect to other networks. For example, if a packet comes in from a client on the Sales VLAN destined for the mail server on ACME's core network, the switch won't know how to treat that packet so it will simply drop it. Junos OS needs to know how to forward this packet to the destination.

ACME has decided that since their network is quite simple, static routes are the easiest way to tell network devices what the next hop is when forwarding traffic to other parts of the network. Whether or not you agree with this, it is what you have to configure. Looking at the IOS switch, you can see that there are two static routes configured, one for the default route and another for the management network:

```
ip route 0.0.0.0 0.0.0.0 192.168.200.1
ip route 10.200.0.0 255.255.255.0 10.100.0.1
```

In Junos OS, static routes are added under the `routing-options` hierarchy. In addition, you must tell Junos OS that this route will be a static route and because static routes in Junos OS are so flexible, it allows you to use options such as `reject`, `discard`, and `qualified next-hop`, so you must specify the keyword `next-hop` before entering the next hop IP address:

```
set routing-options static route 0.0.0.0/0 next-hop 192.168.200.1
set routing-options static route 10.200.0.0/24 next-hop 10.100.0.1
```

The usual command to show routes installed into the routing table would be `show route`, however as interfaces have still not been assigned to the new VLANs, the RVIs are down so no static routes would appear in the routing table.

```
{master:0}
admin@ACME-Juniper-SW-10> show route
```

77 Configuring Spanning Tree Protocol

```
inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
10.100.0.10/32    *[Local/0] 00:39:26
                  Reject
192.168.10.10/32  *[Local/0] 00:39:26
                  Reject
192.168.20.10/32  *[Local/0] 00:39:26
                  Reject
192.168.30.10/32  *[Local/0] 00:39:26
                  Reject
192.168.200.10/32 *[Local/0] 00:39:26
                  Reject
```

Note that for the local routes for the RVIs, the routes are set with a next hop of `Reject`. Once the VLANs are assigned to active interfaces, `Reject` would change to the interface and an additional route of `Direct` would be added, for example, for VLAN 200, the route would look similar to the following:

```
192.168.200.0/24  *[Direct/0] 00:35:42
                  > via vlan.200
192.168.200.10/32 *[Local/0] 00:36:01
                  Local via vlan.200
```

Therefore, in order to confirm that the configuration for the static routes is correct, we could use the command `show configuration routing-options`:

```
{master:0}
admin@ACME-Juniper-SW-10> show configuration routing-options
static {
    route 0.0.0.0/0 next-hop 192.168.200.1;
    route 10.200.0.0/24 next-hop 10.100.0.1;
}
```

These appear to be correct, matching the existing IOS routes, therefore our next task is to move back to Layer 2 and configure the Spanning Tree Protocol.

Configuring Spanning Tree Protocol

At the moment, the designated Catalyst switch has been set with a priority of 8192. The switch at the aggregation layer has been set with a priority of 4092 and switch SW-11 has a priority of 12288.

As illustrated in Figure 7.3, this means that the aggregation switch would be the root bridge for VLANs 100 and 200, however, as VLANs 10, 20, and 30 are only locally significant to switches SW-10 and SW-11, the switch you are replacing would be the root bridge.

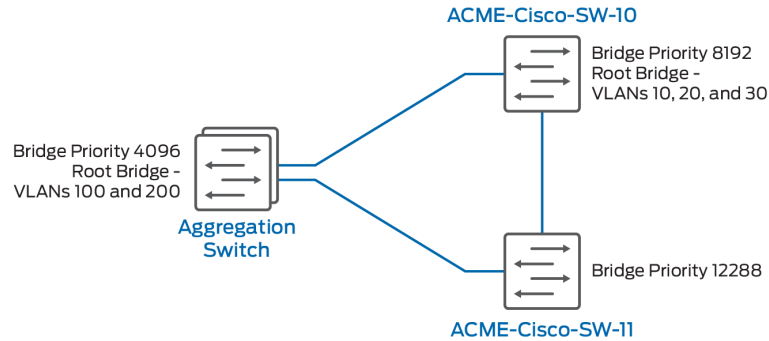


Figure 7.3 Spanning Tree Root Bridges

By default, Cisco Catalyst switches are configured with a version of spanning tree known as “Per VLAN Spanning Tree +”. This is similar to the original version of spanning protocol where by default the ports could take up to 50 seconds to move to the forwarding state. The network administrator configured the switch to use the updated version “Rapid Per VLAN Spanning Tree +” where ports move into the forward state much more quickly. The Cisco Catalyst switch is therefore configured as follows:

```
spanning-tree mode rapid-pvst
spanning-tree vlan 1-4094 priority 8192
```

By comparison, Junos OS switches are configured as “Rapid Spanning Tree Protocol” by default. This means that there is no need to change the existing protocol, all we need to do is set the `bridge-priority` to 4096. In Junos OS, instead of having to set the spanning tree priority in multiples of 4096, Juniper made it easier for network engineers, as all you need to enter is 4k, 8k, 12k, and so on. The command to set our switch with a bridge priority of 8192 is therefore:

```
set protocols rstp bridge-priority 8k
```

Following a commit, the `show spanning-tree bridge` command will confirm that you have set the priority correctly:

```
{master:0}
admin@ACME-Juniper-SW-10> show spanning-tree bridge

STP bridge parameters
Context ID                : 0
Enabled protocol          : RSTP
Root ID                   : 8192.50:c5:8d:a6:98:41
Hello time                 : 2 seconds
Maximum age               : 20 seconds
Forward delay             : 15 seconds
Message age               : 0
Number of topology changes : 0
Local parameters
```

```
Bridge ID           : 8192.50:c5:8d:a6:98:41
Extended system ID  : 0
Internal instance ID : 0
```

NOTE As this switch is being deployed without being connected to the network, the Spanning Tree Protocol will report this switch as being the root bridge. Once the switch is connected to the live network, the Spanning Tree Protocol will correctly report the aggregation switch as being the root bridge.

Paying attention to detail and ensuring VLANs, Routed Virtual Interfaces, and Spanning Tree Protocol are all configured correctly, and that static routes are accurate and point to the right next hop, makes your job much easier as you move to the next stages, which include configuring the interfaces, and hopefully, will also mean less time spent troubleshooting.

MORE? For lots of great details and tutorials on the EX Series, see: *Day One: EX Series Up and Running* at <https://www.juniper.net/us/en/training/jnbooks/day-one/fabric-switching-tech-series/ex-series-up-running/>.

Chapter 8

Link Aggregation

Connecting switches together using two links to provide redundancy, in case an interface or cable breaks, is a great idea. It can also be wasteful as the Spanning Tree Protocol will put one of the interfaces into a blocking state in order to prevent a bridging loop.

So combining two physical links into a single logical link is an excellent alternative, because it offers two additional benefits over having two single links without sacrificing redundancy. The first benefit is that there is no need for the Spanning Tree Protocol to put one of the interface in the blocking state, and the second is that it doubles the bandwidth of the inter-switch links because both interfaces are used to send and receive traffic.

This method of combining two physical interfaces into one logical interface is more commonly known as “link aggregation,” or “port channels” in Cisco language. This method can also be referred to by its standard, which is 802.3ad.

In any case, no matter what you prefer to call link aggregation, in this book’s ACME scenario it is being used to provide inter-switch links. SW-10 and SW-11 are linked together with an aggregated link, SW-10 is connected to the aggregation layer using an aggregated link, and SW-11 is again connected to the aggregation layer using the same type of link.

The 803.2ad standard allows up to sixteen interfaces to be added to a single logical link with eight of those interfaces active at any one time. ACME has just two physical interfaces in each logical link. On SW-10, ACME is using interfaces GigabitEthernet1/1 and GigabitEthernet1/2 as part of port-channel 1, and ports GigabitEthernet1/3 and GigabitEthernet1/4 as part of port-channel 2. Figure 8.1 illustrates ACME’s aggregated links.

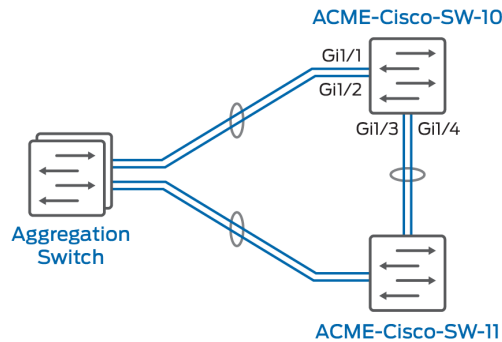


Figure 8.1 ACME SW-10 Link Aggregation

The first step when configuring port channels is to tell the switch which interfaces are to be included in what port channel. In IOS, it is simply a matter of going under each interface and entering the command `channel-group x`, where `x` equals the port number you wish to use, then specifying the mode you want to use if you don't want to use the default mode.

IOS supports two link aggregation protocols, PAgP and LACP, in addition to a third method that forces the aggregated links into an up state, regardless of whether the neighboring switch interfaces are configured as aggregated links or not. Junos OS does support the LACP protocol and the "forced on" configuration, however, because PAgP is Cisco proprietary, Junos OS does not support this protocol. Happily, ACME has set their port channels to use active mode, and this tells IOS to use the protocol LACP to negotiate the use of the port channel with the neighbor. The configuration for port channel 1 therefore looks like this:

```
interface GigabitEthernet1/1
  description <<PORT CHANNEL 1>>
  channel-group 1 mode active
!
interface GigabitEthernet1/2
  description <<PORT CHANNEL 1>>
  channel-group 1 mode active
```

Looking at this configuration, you may notice that there is a description set under the interface. This is because the engineer who originally configured the switch added descriptions to the interface configuration so that other engineers who looked at the switch later on could see what the interface was used for quickly and easily without having to trace cables. The configuration for the interfaces connected to port channel 2 show the description is set to match the `channel-group` setting:

```
interface GigabitEthernet1/3
  description <<PORT CHANNEL 2>>
  channel-group 2 mode active
!
interface GigabitEthernet1/4
```

```
description <<PORT CHANNEL 2>>
channel-group 2 mode active
```

In Junos OS an additional step is required when compared with IOS, and that is telling Junos OS how many aggregated links are going to be configured on the EX Series switch. This option is set as part of the `chassis` hierarchy. In this case, you only need to create two aggregated links, and as such, you specify the `device-count` as 2:

```
set chassis aggregated-devices ethernet device-count 2
```

If you enter this command and then commit the change, Junos OS will add two additional interfaces to the interface list, so if you enter the command `show interfaces terse | match ae`, you will see interfaces `ae0` and `ae1`, which is short for “aggregated ethernet”, in the list of interfaces:

```
{master:0}
admin@ACME-Juniper-SW-10> show interfaces terse | match ae
ae0                up      down
ae1                up      down
```

Specifying the Aggregation Protocol

By default, Junos OS sets the aggregated links to be *forced on*. If you recall, ACME is using LACP, therefore you should set the EX Series switch to use the same protocol. Failure to do so could cause a neighboring switch using LACP to put its interfaces into the suspended state:

```
set interfaces ae0 aggregated-ether-options lacp active
set interfaces ae1 aggregated-ether-options lacp active
```

Now that you have confirmed that the interfaces `ae0.0` and `ae1.0` have been created and they are set to negotiate aggregated links using the LACP protocol, you can assign the relevant physical interfaces to these aggregated interfaces. On the Catalyst, interfaces `GigabitEthernet1/1` and `GigabitEthernet1/2` were assigned to port channel 1, on the EX Series switch, this equates to interfaces `ge-0/1/0` and `ge-0/1/1`. The `ether-options` command followed by `802.3ad` tells Junos that this interface is part of an aggregated link, while `ae0` tells Junos which aggregated link these interfaces are part of, as shown here:

```
set interfaces ge-0/1/0 description <<AGGREGATED-ETHERNET-0>>
set interfaces ge-0/1/0 ether-options 802.3ad ae0
set interfaces ge-0/1/1 description <<AGGREGATED-ETHERNET-0>>
set interfaces ge-0/1/1 ether-options 802.3ad ae0
```

You may have noticed that the interface description has been changed compared to the Catalyst switch configuration. This change is simply to reflect how Junos refers to port channels. The second port channel on the Catalyst includes

interfaces GigabitEthernet1/3 and GigabitEthernet1/4. On the EX Series switch, these translate into interfaces ge-0/1/2 and ge-0/1/3. Both of these interfaces will be added to aggregated ethernet interface 1:

```
set interfaces ge-0/1/2 description <<AGGREGATED-ETHERNET-1>>
set interfaces ge-0/1/2 ether-options 802.3ad ae1
set interfaces ge-0/1/3 description <<AGGREGATED-ETHERNET-1>>
set interfaces ge-0/1/3 ether-options 802.3ad ae1
```

Before you can commit these changes, you need to delete the logical interface unit 0 from interfaces ge-0/1/0, ge-0/1/1, ge-0/1/2, and ge-0/1/3, otherwise the Junos OS will display an error when you try to commit:

```
{master:0}[edit]
admin@ACME-Juniper-SW-10# commit
[edit interfaces ge-0/1/0]
<unit 0>
logical unit is not allowed on aggregated links
error: DCD Configuration check FAILED.
error: configuration check-out failed
```

In order to prevent this error from occurring, you need to delete the logical interfaces using the delete command:

```
delete interfaces ge-0/1/0 unit 0
delete interfaces ge-0/1/1 unit 0
delete interfaces ge-0/1/2 unit 0
delete interfaces ge-0/1/3 unit 0
```

Checking the Configuration

After a successful commit, you can typically check which interfaces are members of the aggregated link by running the show interface ae0 extensive | find Aggregate command or the show interface ae1 extensive | find Aggregate command. The EX Series switch has been configured to use the uplink interfaces ge-0/1/0, ge-0/1/1, ge-0/1/2, and ge-0/1/3. In our case, new SFPs were purchased at the same time as the switch, and these have been installed; this means the output for interface ae0 shows that two interfaces are members of this aggregated link:

```
{master:0}
admin@ACME-Juniper-SW-10> show interfaces ae0 extensive | find Aggregate
```

Aggregate member links: 2

LACP info:	Role	System priority	System identifier	Port priority	Port number	Port key
ge-0/1/1.0	Actor	127	50:c5:8d:a6:98:40	127	2	1
ge-0/1/1.0	Partner	1	00:00:00:00:00:00	1	2	1
ge-0/1/0.0	Actor	127	50:c5:8d:a6:98:40	127	1	1
ge-0/1/0.0	Partner	1	00:00:00:00:00:00	1	1	1

LACP Statistics:	LACP Rx	LACP Tx	Unknown Rx	Illegal Rx
ge-0/1/1.0	0	0	0	0
ge-0/1/0.0	0	0	0	0

```
Marker Statistics:  Marker Rx      Resp Tx      Unknown Rx      Illegal Rx
ge-0/1/1.0          0            0            0            0
ge-0/1/0.0          0            0            0            0
Protocol eth-switch, Generation: 151, Route table: 0
Flags: Is-Primary, Trunk-Mode
```

Running the same command on interface ae1 also confirms that interfaces ge-0/1/2 and ge-0/1/3 are members of aggregated link ae1:

```
{master:0}
admin@ACME-Juniper-SW-10> show interfaces ae1 extensive | find Aggregate
```

```
Aggregate member links: 2
```

```
LACP info:          Role      System      System      Port      Port      Port
                  priority  identifier  priority  number    key
ge-0/1/3.0         Actor      127        50:c5:8d:a6:98:40  127        4        2
ge-0/1/3.0         Partner    1          00:00:00:00:00:00  1          4        2
ge-0/1/2.0         Actor      127        50:c5:8d:a6:98:40  127        3        2
ge-0/1/2.0         Partner    1          00:00:00:00:00:00  1          3        2
LACP Statistics:    LACP Rx      LACP Tx      Unknown Rx      Illegal Rx
ge-0/1/3.0          0            155          0            0
ge-0/1/2.0          0            0            0            0
Marker Statistics:  Marker Rx      Resp Tx      Unknown Rx      Illegal Rx
ge-0/1/3.0          0            0            0            0
ge-0/1/2.0          0            0            0            0
Protocol eth-switch, Generation: 152, Route table: 0
Flags: Trunk-Mode
```

As you can see, the interfaces are part of the aggregated links and they are set to use the LACP protocol, therefore the next task is to tell Junos which interfaces, including ae0 and ae1, are trunk links, which are access ports, and which VLANs they are members of. That's exactly what is covered in Chapter 9.

Chapter 9

Access to Trunk Ports

This means if the switch was connected to SW-11 and the aggregation switch, the EX switch would not understand the frames that were being forwarded to it, which in turn means if the clients and servers were connected, the traffic wouldn't have anywhere to go. In order for this migration to be successful, all of the physical interfaces need to be set as one of three things:

- An access port
- A trunk link
- Shutdowned and disabled

Access ports are basically connected to clients, printers, and so on, and carry data from a single VLAN. Trunk links are usually connected to other switches, occasionally to routers and firewalls, and now, increasingly, to virtual server hosts. In this environment there are two such servers, and they are connected to both SW-10 and SW-11 as per Figure 9.1.

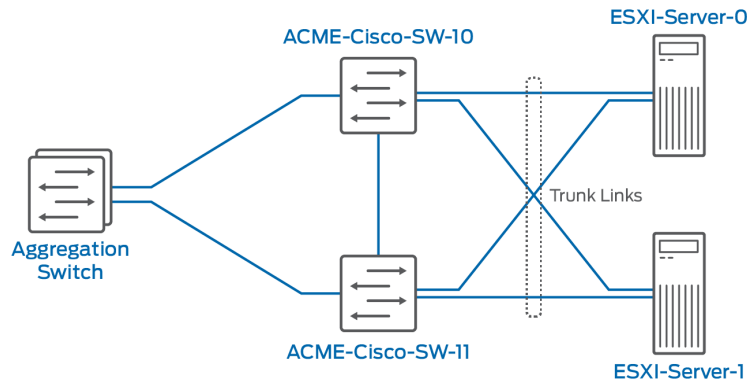


Figure 9.1 Virtual Server Host Connectivity

While it is unusual for servers to be connected to the access layer of a network, ACME has nonetheless done just that, and although it is probably a good idea to raise this issue at the next progress meeting, the task right now is to ensure that all interfaces are not only in the correct mode but also have the correct VLANs assigned, and the first step will be to configure the access ports.

Configuring Access Ports

ACME has seven workstations connected to SW-10. Three workstations are in the Sales VLAN, two workstations are in the Finance VLAN, and the other two are in the Engineering VLAN. In IOS, interfaces are automatically set as access ports, therefore there are only two commands required on these interfaces: the description and the VLAN number:

```

interface GigabitEthernet0/3
  description <<WORKSTATION1>>
  switchport access vlan 10
!
interface GigabitEthernet0/4
  description <<WORKSTATION2>>
  switchport access vlan 10
!
interface GigabitEthernet0/5
  description <<WORKSTATION3>>
  switchport access vlan 10

```

Junos OS is different from IOS in that Junos creates a logical interface under the physical interface by default even if traffic from only one VLAN is being sent out of this interface. The logical interface created by default is unit 0. Therefore, when we use a set command in an interface, we should also include unit 0 otherwise the command will fail.

Interfaces on an EX Series switch can also be routed ports and used for other routed protocols such as CLNS and IPv6, therefore, following unit 0, you should also include the keyword `family`, and for Layer 2 ports, the optional `ethernet-switching`, after which you specify the `vlan members` command followed by the VLAN number this interface is part of, which for workstations 1, 2, and 3, is VLAN 10. The Junos configuration looks like this:

```
set interfaces ge-0/0/2 unit 0 description <<WORKSTATION1>>
set interfaces ge-0/0/2 unit 0 family ethernet-switching vlan members 10
set interfaces ge-0/0/3 unit 0 description <<WORKSTATION2>>
set interfaces ge-0/0/3 unit 0 family ethernet-switching vlan members 10
set interfaces ge-0/0/4 unit 0 description <<WORKSTATION3>>
set interfaces ge-0/0/4 unit 0 family ethernet-switching vlan members 10
```

Workstations 4 and 5 are part of the VLAN Finance, VLAN number 20:

```
interface GigabitEthernet0/6
description <<WORKSTATION4>>
switchport access vlan 20
!
interface GigabitEthernet0/7
description <<WORKSTATION5>>
switchport access vlan 20
```

So that means the equivalent Junos OS command is as follows:

```
set interfaces ge-0/0/5 unit 0 description <<WORKSTATION4>>
set interfaces ge-0/0/5 unit 0 family ethernet-switching vlan members 20
set interfaces ge-0/0/6 unit 0 description <<WORKSTATION5>>
set interfaces ge-0/0/6 unit 0 family ethernet-switching vlan members 20
```

Finally, workstations 6 and 7 are part of the Engineering VLAN, which is VLAN number 30:

```
interface GigabitEthernet0/8
description <<WORKSTATION6>>
switchport access vlan 30
!
interface GigabitEthernet0/9
description <<WORKSTATION7>>
switchport access vlan 30
```

Therefore, we use the following to add the EX Series switch interfaces `ge-0/0/7` and `ge-0/0/8` do the same VLAN:

```
set interfaces ge-0/0/7 unit 0 description <<WORKSTATION6>>
set interfaces ge-0/0/7 unit 0 family ethernet-switching vlan members 30
set interfaces ge-0/0/8 unit 0 description <<WORKSTATION7>>
set interfaces ge-0/0/8 unit 0 family ethernet-switching vlan members 30
```

Configuring Trunk Links

The next step is to configure the trunk links. The ACME switch has four trunk links in total: one to SW-11, one to the aggregation switch, and two to the virtual server hosts. The first interfaces you are going to configure are the links to the servers. There are several items of importance in the IOS configuration that you need to be aware of.

The first thing you need to be aware of is the command `switchport trunk encapsulation dot1q`, which tells IOS to use a protocol called 802.1Q to tag frames with the VLAN number to which that frame belongs. This is essentially how trunk links are able to carry data from multiple VLANs. The reason why it is included is because IOS also supports another VLAN tagging protocol called “ISL”. Junos OS only supports the protocol 802.1Q, therefore there is no equivalent command in Junos OS.

The second thing you need to take into account is that within the IOS configuration is the command `switchport trunk native vlan 900`. This is a security mechanism to prevent what is known as VLAN hopping. Finally, it is important to note that both trunk links are configured to allow traffic from VLANs 10, 20, and 30. This is obviously to allow local connectivity from the three local VLANs:

```
interface GigabitEthernet0/1
description <<ESXI-SERVER-01>>
switchport trunk allowed vlan 10,20,30
switchport trunk encapsulation dot1q
switchport trunk native vlan 900
switchport mode trunk
!
interface GigabitEthernet0/2
description <<ESXI-SERVER-02>>
switchport trunk allowed vlan 10,20,30
switchport trunk encapsulation dot1q
switchport trunk native vlan 900
switchport mode trunk
```

Looking at the method of adding VLANs to Junos OS trunk links, there are two ways in which you can achieve this. One way would be to enter the allowed VLANs one at a time:

```
set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members 10
set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members 20
set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members 30
set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members 100
set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members 200
```

Entering these commands this way however, can be a bit time-consuming, especially if there is a range of 100 VLANs to be entered. The preferred way of entering a list of VLANs is to include all allowed VLANs within a single command, but encapsulated in square brackets. This can include separate VLANs:

```
set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members [ 10 20 30 ]
```

Alternatively, the command can include a range of VLANs or a mixture of the two:

```
set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members [ 10-30 100]
```

Therefore, when configuring the interfaces on the EX Series switch, you will tell Junos OS that these interfaces are trunk links by using the `port-mode trunk` command. You can also tell Junos that the native VLAN ID is 900 and then add the list of allowed VLANs by using the method discussed before, like this:

```
set interfaces ge-0/0/0 unit 0 description <<ESXI-SERVER-01>>
set interfaces ge-0/0/0 unit 0 family ethernet-switching port-mode trunk
set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members [ 10 20 30 ]
set interfaces ge-0/0/0 unit 0 family ethernet-switching native-vlan-id 900

set interfaces ge-0/0/1 unit 0 description <<ESXI-SERVER-02>>
set interfaces ge-0/0/1 unit 0 family ethernet-switching port-mode trunk
set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members [ 10 20 30 ]
set interfaces ge-0/0/1 unit 0 family ethernet-switching native-vlan-id 900
```

Once the interfaces connected to the servers have been configured, the next trunk links to be configured are those being used as switch uplinks to SW-11 and the aggregation switch. These are different from the interfaces connected to the servers. These interfaces are not physical interfaces, and on the Catalyst switch, these are port channel interfaces.

That said, however, the configuration is very similar to the physical interface configuration in that the encapsulation is set and the native VLAN is set to 900. The only real difference is that these trunk links allow different VLANs to transit the links: VLANs 100 and 200 to the aggregation switch and VLANs 10, 20, 30, 100, and 200 to switch SW-11:

```
interface Port-channel1
description <<UPLINK_TO_AGG_LAYER>>
switchport trunk allowed vlan 100,200
switchport trunk encapsulation dot1q
switchport trunk native vlan 900
switchport mode trunk
!
interface Port-channel2
description <<UPLINK_TO_SW-11>>
switchport trunk allowed vlan 10,20,30,100,200
switchport trunk encapsulation dot1q
switchport trunk native vlan 900
switchport mode trunk
```

On the EX Series switch, you are again not configuring physical interfaces but instead you are configuring aggregated Ethernet interfaces. The configuration under the `ae0` and `ae1` interfaces is very similar to how `ge-0/0/0` and `ge-0/0/1` were configured with the exception of the VLANs that the links are members of:

```
set interfaces ae0 unit 0 description <<UPLINK_TO_AGG_LAYER>>
set interfaces ae0 unit 0 family ethernet-switching port-mode trunk
set interfaces ae0 unit 0 family ethernet-switching vlan members [ 100 200 ]
set interfaces ae0 unit 0 family ethernet-switching native-vlan-id 900

set interfaces ae1 unit 0 description <<UPLINK_TO_SW-11>>
set interfaces ae1 unit 0 family ethernet-switching port-mode trunk
set interfaces ae1 unit 0 family ethernet-switching vlan members [ 10 20 30 100 200 ]
set interfaces ae1 unit 0 family ethernet-switching native-vlan-id 900
```

Disabling Spare Interfaces

Now that all the interfaces to be connected to clients, servers, and other switches have been configured, the question to ask is: what should you do with the interfaces that are spare? You could just leave them as they are, however, this is not recommended. As a security best practice, both Juniper Networks and Cisco recommend that all interfaces not being used should be both shut down and placed into the VLAN that is being used to set the native VLAN ID on the trunk links.

Following the security best practice guidelines, interfaces GigabitEthernet0/10 through GigabitEthernet0/24 on the Cisco switch have been shut down and interfaces have been made members of VLAN 900. In addition to this, the description of all of these interfaces have been set to <<UNUSED>> to assist any engineer who supports this switch at a later date:

```
interface GigabitEthernet0/10
description <<UNUSED>>
switchport access vlan 900
shutdown
!
interface GigabitEthernet0/11
description <<UNUSED>>
switchport access vlan 900
shutdown
!
interface GigabitEthernet0/12
description <<UNUSED>>
switchport access vlan 900
shutdown
!
interface GigabitEthernet0/13
description <<UNUSED>>
switchport access vlan 900
shutdown
!
interface GigabitEthernet0/14
description <<UNUSED>>
switchport access vlan 900
shutdown
!
interface GigabitEthernet0/15
description <<UNUSED>>
switchport access vlan 900
```

```

shutdown
!
interface GigabitEthernet0/16
description <<UNUSED>>
switchport access vlan 900
shutdown
!
interface GigabitEthernet0/17
description <<UNUSED>>
switchport access vlan 900
shutdown
!
interface GigabitEthernet0/18
description <<UNUSED>>
switchport access vlan 900
shutdown
!
interface GigabitEthernet0/19
description <<UNUSED>>
switchport access vlan 900
shutdown
!
interface GigabitEthernet0/20
description <<UNUSED>>
switchport access vlan 900
shutdown
!
interface GigabitEthernet0/21
description <<UNUSED>>
switchport access vlan 900
shutdown
!
interface GigabitEthernet0/22
description <<UNUSED>>
switchport access vlan 900
shutdown
!
interface GigabitEthernet0/23
description <<UNUSED>>
switchport access vlan 900
shutdown
!
interface GigabitEthernet0/24
description <<UNUSED>>
switchport access vlan 900
shutdown

```

When the Catalyst switch was originally configured, chances are the engineer who configured it used the `interface range` command so the same configuration could be set on multiple interfaces at the same time without having to enter the same configuration on 15 interfaces, 15 times.

The Junos OS also has a similar command, which can be used to simultaneously set the same configuration on multiple interfaces. What's different about Junos OS is the `interface-range` command is used to create a group. The interfaces are then assigned to the group along with the settings you would like interfaces assigned to

that group to use.

The first step in creating an interface range is to set the name of the group and then specify the interfaces or range of interfaces. In our case, the group will be named `UNUSED`. The option `member-range` will be used to specify the range of interfaces that belong to this group. This range is from interfaces `ge-0/0/9` to `ge-0/0/23`:

```
set interfaces interface-range UNUSED member-range ge-0/0/9 to ge-0/0/23
```

The commands to set the description, the VLANs the interfaces are members of, and that the interfaces are disabled, are exactly the same as they would be under the interfaces themselves, except, instead of saying `set interface ge-0/0/x` and then the command, with interface ranges, you specify `set interface interface-name <group-name>` and then use exactly the same commands:

```
set interfaces interface-range UNUSED disable
set interfaces interface-range UNUSED unit 0 description <<UNUSED>>
set interfaces interface-range UNUSED unit 0 family ethernet-switching vlan members UNUSED
```

All that remains is to commit the configuration, and then, assuming there are no errors, use various `show` commands to check that the configuration is correct and matches the device it is to replace.

Sanity Checking the Configuration

You may have noticed that every interface you have configured has been given a description, and as mentioned, this description is to assist future engineers who may need to support the switch in showing what device is connected to which interface, and to show which interfaces are spare. The `show interfaces descriptions` command lists all interfaces on the switch, including aggregated Ethernet links and the assigned description, along with whether the interface is up or down:

```
{master:0}
admin@ACME-Juniper-SW-10> show interfaces descriptions
Interface      Admin Link Description
ge-0/0/0.0      up    down <<ESXI-SERVER-01>>
ge-0/0/1.0      up    down <<ESXI-SERVER-02>>
ge-0/0/2.0      up    down <<WORKSTATION1>>
ge-0/0/3.0      up    down <<WORKSTATION2>>
ge-0/0/4.0      up    down <<WORKSTATION3>>
ge-0/0/5.0      up    down <<WORKSTATION4>>
ge-0/0/6.0      up    down <<WORKSTATION5>>
ge-0/0/7.0      up    down <<WORKSTATION6>>
ge-0/0/8.0      up    down <<WORKSTATION7>>
ge-0/0/9.0      up    down <<UNUSED>>
ge-0/0/10.0     up    down <<UNUSED>>
ge-0/0/11.0     up    down <<UNUSED>>
ge-0/0/12.0     up    down <<UNUSED>>
ge-0/0/13.0     up    down <<UNUSED>>
ge-0/0/14.0     up    down <<UNUSED>>
ge-0/0/15.0     up    down <<UNUSED>>
ge-0/0/16.0     up    down <<UNUSED>>
```

93 Sanity Checking the Configuration

```

ge-0/0/17.0    up    down <<UNUSED>>
ge-0/0/18.0    up    down <<UNUSED>>
ge-0/0/19.0    up    down <<UNUSED>>
ge-0/0/20.0    up    down <<UNUSED>>
ge-0/0/21.0    up    down <<UNUSED>>
ge-0/0/22.0    up    down <<UNUSED>>
ge-0/0/23.0    up    down <<UNUSED>>
ge-0/1/0       up    down <<AGGREGATED-ETHERNET-0>>
ge-0/1/1       up    down <<AGGREGATED-ETHERNET-0>>
ge-0/1/2       up    down <<AGGREGATED-ETHERNET-1>>
ge-0/1/3       up    down <<AGGREGATED-ETHERNET-1>>
ae0.0          up    down <<UPLINK_TO_AGG_LAYER>>
ae1.0          up    down <<UPLINK_TO_SW-11>>

```

Next, you should ensure that all VLANs have been created and that all interfaces have been made members of the VLANs you have created. Under no circumstances should any interface be a member of the default VLAN. The `show vlans` command lists each VLAN and next to the VLAN name is the VLAN number and which interfaces are members of that VLAN. Any interface with an asterisk next to it is up. On our switch, all of the ports are currently down.

```

{master:0}
admin@ACME-Juniper-SW-10> show vlans
Name          Tag    Interfaces
ENGINEERING   30     ae1.0, ge-0/0/0.0, ge-0/0/1.0, ge-0/0/7.0, ge-0/0/8.0
FINANCE       10     ae1.0, ge-0/0/0.0, ge-0/0/1.0, ge-0/0/2.0, ge-0/0/3.0,
          ge-0/0/4.0
MANAGEMENT    100    ae0.0, ae1.0
SALES         20     ae1.0, ge-0/0/0.0, ge-0/0/1.0, ge-0/0/5.0, ge-0/0/6.0
TRANSIT       200    ae0.0, ae1.0
UNUSED        900    ae0.0, ae1.0, ge-0/0/0.0, ge-0/0/1.0, ge-0/0/9.0,
          ge-0/0/10.0, ge-0/0/11.0, ge-0/0/12.0, ge-0/0/13.0,
          ge-0/0/14.0, ge-0/0/15.0, ge-0/0/16.0, ge-0/0/17.0,
          ge-0/0/18.0, ge-0/0/19.0, ge-0/0/20.0, ge-0/0/21.0,
          ge-0/0/22.0, ge-0/0/23.0
default       None

```

The next thing you should check is that the interfaces that should be set as trunk links are in fact, trunk links, and you've not inadvertently configured an interface as a trunk link that should be an access port. The `show ethernet-switching interfaces detail` command will show you whether a port is a trunk link or an access port, however, this will output a lot of information. If we instead filter using the `| match Trunk` option, only the line which states the port mode is a trunk will be displayed along with the interface:

```

{master:0}
admin@ACME-Juniper-SW-10> show ethernet-switching interfaces detail | match Trunk

```

```
Interface: ae0.0, Index: 69, State: down, Port mode: Trunk
Interface: ae1.0, Index: 70, State: down, Port mode: Trunk
Interface: ge-0/0/0.0, Index: 78, State: down, Port mode: Trunk
Interface: ge-0/0/1.0, Index: 79, State: down, Port mode: Trunk
```

The remaining interfaces obviously aren't trunk links, and that being the case, the only other thing they can be are either access ports or members of aggregated links. The Catalyst switch has interfaces GigabitEthernet0/3 to GigabitEthernet0/24 configured as access ports. On the EX Series switch this translates to interfaces ge-0/0/2 to ge-0/0/23. In order to confirm that these ports are configured correctly, you can use the same `show ethernet-switching interfaces detail` command but instead set the filter to `| match Access`:

```
{master:0}
admin@ACME-Juniper-SW-10> show ethernet-switching interfaces detail | match Access
Interface: ge-0/0/2.0, Index: 80, State: down, Port mode: Access
Interface: ge-0/0/3.0, Index: 81, State: down, Port mode: Access
Interface: ge-0/0/4.0, Index: 82, State: down, Port mode: Access
Interface: ge-0/0/5.0, Index: 83, State: down, Port mode: Access
Interface: ge-0/0/6.0, Index: 84, State: down, Port mode: Access
Interface: ge-0/0/7.0, Index: 85, State: down, Port mode: Access
Interface: ge-0/0/8.0, Index: 86, State: down, Port mode: Access
Interface: ge-0/0/9.0, Index: 87, State: down, Port mode: Access
Interface: ge-0/0/10.0, Index: 88, State: down, Port mode: Access
Interface: ge-0/0/11.0, Index: 89, State: down, Port mode: Access
Interface: ge-0/0/12.0, Index: 90, State: down, Port mode: Access
Interface: ge-0/0/13.0, Index: 91, State: down, Port mode: Access
Interface: ge-0/0/14.0, Index: 92, State: down, Port mode: Access
Interface: ge-0/0/15.0, Index: 93, State: down, Port mode: Access
Interface: ge-0/0/16.0, Index: 94, State: down, Port mode: Access
Interface: ge-0/0/17.0, Index: 95, State: down, Port mode: Access
Interface: ge-0/0/18.0, Index: 96, State: down, Port mode: Access
Interface: ge-0/0/19.0, Index: 97, State: down, Port mode: Access
Interface: ge-0/0/20.0, Index: 98, State: down, Port mode: Access
Interface: ge-0/0/21.0, Index: 99, State: down, Port mode: Access
Interface: ge-0/0/22.0, Index: 100, State: down, Port mode: Access
Interface: ge-0/0/23.0, Index: 101, State: down, Port mode: Access
```

If you are confident that the interfaces are part of the correct VLAN, that the trunk interfaces are set to allow the correct VLANs to traverse them, and that the spare interfaces have been disabled and are part of the holding VLAN, the next thing to do is to think about redundancy, otherwise a link failure has the potential to prevent clients from accessing network resources. This is exactly what we look at in Chapter 10.

Chapter 10

Redundancy

Building redundancy into your switching infrastructure can not only help prevent outages due to software and hardware failures, but can also help minimize maintenance windows by ensuring continuity of service.

Juniper provides a number of mechanisms within the EX Series platform to help you ensure resiliency within your network, with two of the most popular being Virtual Router Redundancy Protocol, more commonly known as VRRP, and Virtual Chassis, or VC.

Virtual Router Redundancy Protocol (VRRP)

VRRP is a networking protocol that provides Layer 3 gateway resiliency. This is typically deployed with a master router with one or more backup routers. Network devices such as clients, servers, printers, and so on, need only to be told the virtual IP address. When the network device sends an ARP request for the virtual IP address, the master router would respond with a virtual MAC address. This virtual MAC address is then entered into the switch's MAC address tables and network device's ARP tables so that all network devices are aware of how to reach this virtual MAC address.

Figure 10.1 shows an example of a small network utilizing VRRP, where one of the routers has an interface address of 192.168.1.2, and the other router has an address of 192.168.1.3. The client can only be configured with a single default gateway, so in this instance, both routers are configured with a virtual IP address of 192.168.1.1, and the client has its default gateway set to this IP address, as opposed to either of the routers physical addresses.

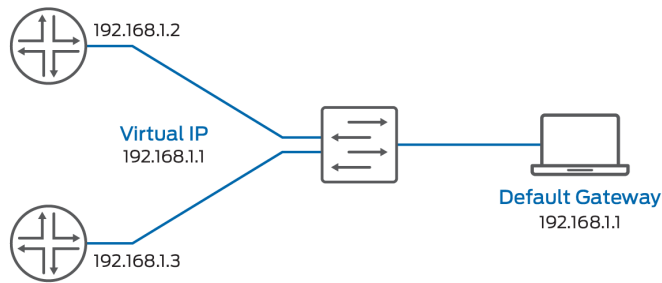


Figure 10.1 VRRP Example

The purpose of the backup routers is to monitor the master router. In the event of a failure of the master router, the backup device will assume the master role and thus take ownership of the virtual address, sending ARP responses to update MAC address tables so that any switches in the path know how to reach this new master router.

As ACME is performing a like-for-like replacement of their access layer, they are mimicking the functionality of their existing Cisco switches. ACME's existing environment is currently utilizing HSRP to provide resiliency of the Layer 3 interfaces. HSRP is very similar to VRRP, with the exception that HSRP is a Cisco proprietary protocol, whereas VRRP is a standard supported by multiple vendors.

The Cisco switch currently utilizes HSRP on four virtual interfaces. The interfaces on the master currently have a priority of 110 set, which is higher than the default priority of 100. The keyword `preempt` is used so that the master will automatically become the master again in the event of a temporary loss of the master router. The virtual IP address is set using the command `standby 10 ip` followed by the IP address.

The existing configuration on ACME's catalyst switch is therefore as follows:

```
interface Vlan10
  standby version 2
  standby 10 ip 192.168.10.1
  standby 10 priority 110
  standby 10 preempt
!
interface Vlan20
  standby version 2
  standby 20 ip 192.168.20.1
  standby 20 priority 110
  standby 20 preempt
!
interface Vlan30
  standby version 2
  standby 30 ip 192.168.30.1
```

97 Virtual Router Redundancy Protocol (VRRP)

```
standby 30 priority 110
standby 30 preempt
!
interface Vlan200
standby version 2
standby 200 ip 192.168.200.12
standby 200 priority 110
standby 200 preempt
!
```

In Junos, the VRRP configuration is also applied under the Layer 3 VLAN interface, more commonly known as an RVI or Routed VLAN Interface. Within each RVI the physical device has its own IP address and VRRP instance. Within the VRRP instance the shared virtual address, priority, and preempt options are set, an example of which can be seen here:

```
unit 10 {
    family inet {
        address 192.168.10.10/24 {
            vrrp-group 10 {
                virtual-address 192.168.10.1;
                priority 110;
                preempt;
            }
        }
    }
}
```

To configure VRRP on an interface you need to define four elements. The first is the VRRP group itself and this component is the most important because within the configuration hierarchy all other elements are subordinate to the VRRP group. The VRRP group number must be consistent across all physical devices as it acts as the identifier for the VRRP instance. If a VRRP group of 35 is assigned to VLAN 35 on the master, then a VRRP group of 35 must also be assigned to VLAN 35 on any backup routers.

To create the VRRP group, you can edit the VRRP group under the interfaces > interface name > unit number > family > inet > address hierarchy:

```
edit interfaces vlan unit 10 family inet address 192.168.10.10/24 vrrp-group 10]
```

Once the VRRP group has been defined you create the virtual address. This is the 'floating address' which is owned by whichever device is the VRRP master:

```
[edit interfaces vlan unit 10 family inet address 192.168.10.10/24 vrrp-group 10]
Set virtual address 192.168.10.1
```

After the virtual address has been set, you can set the VRRP priority of the device. As stated previously, this priority determines which device is the master. The priority is an integer between 1 and 254 and the device with the highest value is the master. In the event that the backup devices were configured with the same priority value, or if no priority was set on any device, the physical device with the highest IP address would be elected as master router and any devices with lower IP addresses would automatically be set as backup devices.

The priority command is therefore set as follows:

```
[edit interfaces vlan unit 10 family inet address 192.168.10.10/24 vrrp-group 10]
Set priority 110
```

The final element to be configured is preempt. This option is actually enabled by default within the Junos OS, which means you don't necessarily need to add it, and, in reality, you would only need to issue a preempt-related command if the default behavior needed to be changed.

NOTE While preempt is a useful feature for ensuring that traffic automatically fails back to the designated master device, it can in certain failure scenarios exacerbate a network issue. For example, if the device with the highest priority is continuously crashing and rebooting this could cause the master role to flap continuously between the routers. This scenario would effectively negate the benefits provided by this protocol.

Should you so wish, you can combine these commands together into a single set command under the interfaces > interface name > unit number > family > inet > address hierarchy:

```
[edit interfaces vlan unit 10 family inet address 192.168.10.10/24]
set vrrp-group 10 virtual address 192.168.10.1 priority 110
```

Therefore, in order to configure VRRP in Junos OS so that the EX switch is similar to the current Catalyst configuration, the following commands should be used:

```
edit interfaces vlan unit 10 family inet address 192.168.10.10/24

set vrrp-group 10 virtual address 192.168.10.1 priority 110
up 4

edit unit 20 family inet address 192.168.20.10/24
set vrrp-group 20 virtual address 192.168.20.1 priority 110
up 4

edit unit 30 family inet address 192.168.30.10/24
set vrrp-group 30 virtual address 192.168.30.1 priority 110
up 4

edit unit 200 family inet address 192.168.200.10/24
set vrrp-group 200 virtual address 192.168.200.12 priority 110
```

Once this configuration has been committed, the `show vrrp` command can be used to verify the configuration is connected and that the ports are up:

```
admin@ACME-Juniper-SW-10> show vrrp
```

Interface	State	Group	VR state	VR Mode	Timer	Type	Address
vlan.10	down	10	init	Active	N	0.000	lcl
							192.168.10.10
vlan.20	down	20	init	Active	N	0.000	vip
							192.168.10.1
vlan.30	down	30	init	Active	N	0.000	lcl
							192.168.20.10
vlan.30	down	30	init	Active	N	0.000	vip
							192.168.20.1
vlan.30	down	30	init	Active	N	0.000	lcl
							192.168.30.10
vlan.30	down	30	init	Active	N	0.000	vip
							192.168.30.1

```
vlan.200    down    200  init Active    N    0.000    lcl    192.168.200.10
                                         vip    192.168.200.12
```

In this case, because the switch is being built, no ports that are part of VLANs 10, 20, 30, and 200 are connected, therefore these RVIs will all be showing as down. At the end of the output you can see the Types `lcl` and `vip`. These indicate whether the address in the Address column is a physical or local address of that interface or whether the address is a Virtual Address or VIP.

Virtual Chassis

As an alternative to using VRRP, Juniper created the Virtual Chassis, which allows the interconnection of multiple switches into a unified single logical device. This is very similar to Cisco's stacking technology, however, with Virtual Chassis you don't need to use special stacking ports. Instead you can choose to use fiber or copper connections to connect switches together in a Virtual Chassis. In addition, almost all Juniper switches support Virtual Chassis, including the low-end EX2200.

Virtual chassis provides a number of benefits over ACME's current VRRP deployment in that it offers simplified management where both switches are managed under a single IP address. While ACME currently has only two switches within the access layer, they could add additional switches, and were that to grow so would the complexity of management. In a Virtual Chassis deployment ACME only has a single logical device to manage.

In addition to simplified management, Virtual Chassis also offers an easy way to expand port capacity. ACME could use Virtual Chassis to pre-provision switches and connect them to the VC in the same manner as a line card would be added to larger, more expensive chassis-based switches.

NOTE There is a limit to how many switches can be in a Virtual Chassis. For example, with EX2200 and EX2300 switches, a maximum of four switches can be part of a single Virtual Chassis, however, with EX3300, you can originally have six switches in a VC. In Junos OS 12.2R1 this limit was increased to ten switches.

A final benefit of using Virtual Chassis is that it also offers resiliency of the RE or Routing Engine. VRRP provides resiliency for Layer 3 interfaces but it is limited in scope to a subnet per instance. In a Virtual Chassis configuration all routing protocol databases, forwarding tables, and configurations are synchronized between the primary and backup routing engines. This means that in the event of a failure the backup routing engine does not need to relearn forwarding information, and maintains network connectivity with no, or at least minimal, disruption.

Figure 10.2 shows how the topology would change if ACME chose to use a Virtual Chassis as opposed to VRRP.

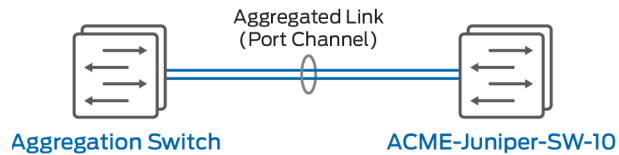


Figure 10.2 Virtual Chassis

You may notice that compared to *Figure P.1, ACME's Network Topology*, on page 62, there is only a single port channel or “aggregated link” between the aggregation switch and ACME-Juniper-SW-10, whereas in their existing Cisco environment ACME has a port channel between ACME-Cisco-SW-10 and the aggregation switch, a second port channel between the aggregation switch and ACME-Cisco-SW-11, and a third port channel between ACME-Cisco-SW-10 and ACME-Cisco-SW-11. Within a Virtual Chassis, aggregated links can span multiple virtual chassis members, reducing the number of interfaces reserved just for the purposes of interconnecting switches.

NOTE The way in which you configure Virtual Chassis on EX Series switches differs between models, therefore learning how to configure Virtual Chassis in this book, and this section, really only serves to make you aware such a technology does exist. Should you wish to learn more about how to configure Virtual Chassis, then visit Juniper's TechLibrary:: https://www.juniper.net/documentation/en_US/junos/topics/task/configuration/virtual-chassis-ex4200-cli.html.

Device redundancy is extremely important, as it ensures minimum down time in the event of a failure, and once you are confident that your new switches provide the same level of redundancy as ACME's existing environment, you can then move on to the next important step, hardening the device security to protect ACME against attackers and intruders.

Chapter 11

Security Hardening

When deploying any device into your network it is important to ensure it is securely configured and cannot be exploited to gain access into your network. But that topic is too huge to tackle here. With this in mind, let's focus on a like-for-like configuration with ACME's Cisco switches.

MORE? There is an entire book, and a very good one, dedicated to hardening Junos devices: *This Week: Hardening Junos Devices, 2nd Edition*, by John Weidley, that is available in the *Day One* library: <https://www.juniper.net/us/en/training/jnbooks/day-one/fundamentals-series/hardening-junos-devices-checklist/>. Be sure to read it to understand all the hardening capabilities of the Junos OS.

Switch Port Security

The existing Cisco switches are configured with the following port security:

```
interface GigabitEthernet0/8
description <<WORKSTATION6>>
switchport access vlan 30
switchport port-security maximum 2
switchport port-security violation restrict
switchport port-security mac-address sticky
```

This configuration sets the port to automatically learn the first two MAC addresses and to prevent any other physical addresses from connecting via the switch port. This configuration has been applied to all access ports. We can implement exactly the same controls and criteria on the EX Series switches, but the commands are simplified. Rather than defining the violation action to restrict, this is a default action within Junos:

```
set interface ge-0/0/8 mac-limit 2
set interface ge-0/0/8 persistent-learning
```

Disabling and placing unused ports into a quarantine VLAN was discussed in Chapter 9, nonetheless, it is still an important part of the security hardening process, and so is worth mentioning again, albeit briefly.

Management Access Restrictions

The Cisco switches use an extended ACL tied to the VTY lines. You can replicate this functionality in a very similar fashion by creating a firewall filter and applying it to the loop back interface. To create the filter, you need to enter the firewall > family > inet hierarchy:

```
edit firewall family inet
```

After which you can create the filter. In this case, the filter will be given the name MAN_FILTER:

```
[edit firewall family inet]
edit filter MAN_FILTER
```

Under each filter you can create “terms”. These are useful as they mean you can easily add more conditions without needing to edit existing conditions. In this case let’s create the term “1”:

```
[edit firewall family inet filter MAN_FILTER]
edit term 1
```

Finally, you can specify which subnets are allowed to manage this switch and to do this specify the source-address and then follow this with an accept statement:

```
[edit firewall filter family inet filter MAN_FILTER term 1]
set from source-address 10.200.0.0/24
set then accept
```

Like Cisco ACLs, firewall filters have an implicit deny at the end, therefore unless you are logging failed access attempts, you can leave the filter as it is and all other traffic will be denied.

Once complete the configuration should look like this:

```
firewall {
  family inet {
    filter MAN_FILTER {
      term 1 {
        from {
          source-address {
            10.200.0.0/24;
          }
        }
        then accept;
      }
    }
  }
}
```

Once you have created the filter, you need to apply it to the loopback interface using the following:

```
set interface lo0 family inet filter input MAN_FILTER
```

Once complete the configuration should look like this:

```
lo0 {
    unit 0 {
        family inet {
            filter {
                input MAN_FILTER;
            }
        }
    }
}
```

Enforce Secure Channels

To prevent compromise through eavesdropping or man-in-the-middle attacks, you should always use secure, encrypted communication channels to administer devices. By default, Junos has SSH enabled and plain text protocols such as Telnet are disabled. To bring the configuration in line with the Cisco switches, you need to disable the Web Management interface on both HTTP and HTTPS. This can be achieved with a single command:

```
Delete system services web-management
```

Set Login Banner

Login banners, while offering no real security against determined and persistent attackers, often act as a deterrent to the more opportunistic. Junos provides a lot of flexibility around creating a login banner, thus preventing users from having to use TABs and spaces to get the formatting correct. You can format the message using the following special characters:

- \n—New line
- \t—Horizontal tab
- \'—Single quotation mark
- \"—Double quotation mark
- \\—Backslash

To enter our login banner as it is from the Cisco switches you would enter the following commands:

```
[Edit system login]
Message "*****\n*
*\n* This is a private network device. If you *\n* are not authorized to connect to this *\n*
device, please disconnect immediately. *\n*
*\n*****"
```

Once committed the above configuration should display our login banner formatted as below:

```
*****
*
* This is a private network device.  If you
*   are not authorized to connect to this
*   device, please disconnect immediately.
*
*****
```

Now all you need to do is perform a final commit and your new EX Series switch should be as secure as the switches they are replacing, if not more so. All that remains is to try to connect to the switch from subnets that are not allowed, or to connect to the switch to make sure access is denied to everyone except authorized personnel using secure protocols by using Telnet or HTTP.

MORE? For a really great book on setting up the EX Series, see: *Day One: EX Series Up and Running* at: <https://www.juniper.net/us/en/training/jnbooks/day-one/fabric-switching-tech-series/ex-series-up-running/>.

Conclusion: The Migration Process

Hopefully, all of your tests have been successful and the device is performing as expected. That being the case, you can now look at bringing this device into service.

This is not simply a matter of unplugging the old switch and plugging the new one in. ACME, like most good companies, has a QA process. This process should include recommended software versions, recommended settings, and should ensure important security features, such as a firewall filter applied to lo0.0 have been configured.

In addition to this, a change plan should be written which includes a detailed plan of the steps that will be taken during the migration from the Catalyst switch to the Ex Series switch. It could be a fairly simple plan like the one here:

Unplug the ethernet cable labeled R.123 from gi0/1 on switch ACME-Cisco-SW-10

Plug ethernet cable labeled R.123 into ge-0/0/0.0 on switch ACME-Juniper-SW-10

Unplug the ethernet cable labeled R.124 from gi0/2 on switch ACME-Cisco-SW-10

Plug ethernet cable labeled R.124 into ge-0/0/1.0 on switch ACME-Juniper-SW-10

Unplug the ethernet cable labeled R.125 from gi0/3 on switch ACME-Cisco-SW-10

Plug ethernet cable labeled R.125 into ge-0/0/2.0 on switch ACME-Juniper-SW-10

.....

Connect to switch ACME-Juniper-SW-10 via SSH

Check ports are showing as up and up

Check for any errors or collisions

.....

As simple as this plan may be, it is none the less important to list every single step along with any tests that must to be performed and in addition a rollback plan should also be written so that every engineer involved with the migration knows exactly what needs to be done next, how they can confirm everything is working as expected, and more importantly, what to do if the migration fails. Of course, if the device has been built according to this book, then a rollback should not be necessary.