Sun™ Secure Application Switch — Configuration and Implementation Guide

Sun Microsystems, Inc.
www.sun.com

Part No. 819-7595-10
October 2006, Revision A

Submit comments about this document at: `http://www.sun.com/hwdocs/feedback`

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at http://www.sun.com/patents and one or more additional patents or pending patent applications in the U.S. and in other countries.

U.S. Government Rights - Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

This distribution may include materials developed by third parties.

Regular expression support is provided by the PCRE library package, which is open source software, written by Philip Hazel, and copyright by the University of Cambridge, England - ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and in other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems and the Sun logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Products covered by and information contained in this service manual are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Sun Microsystems, Inc. détient les droits de propriété intellectuels relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plus des brevets américains listés à l'adresse http://www.sun.com/patents et un ou les brevets supplémentaires ou les applications de brevet en attente aux Etats - Unis et dans les autres pays.

Cette distribution peut comprendre des composants développés par des tierces parties.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems et le logo Sun sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays.

Les produits qui font l'objet de ce manuel d'entretien et les informations qu'il contient sont regis par la legislation americaine en matiere de controle des exportations et peuvent etre soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucleaires, des missiles, des armes biologiques et chimiques ou du nucleaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou reexportations vers des pays sous embargo des Etats-Unis, ou vers des entites figurant sur les listes d'exclusion d'exportation americaines, y compris, mais de maniere non exclusive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une facon directe ou indirecte, aux exportations des produits ou des services qui sont regi par la legislation americaine en matiere de controle des exportations et la liste de ressortissants specifiquement designes, sont rigoureusement interdites.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFACON.

Please
Recycle

Adobe PostScript

Regulatory Compliance Statements

Your Sun product is marked to indicate its compliance class:

- Federal Communications Commission (FCC) — USA
- Industry Canada Equipment Standard for Digital Equipment (ICES-003) — Canada
- Voluntary Control Council for Interference (VCCI) — Japan
- Bureau of Standards Metrology and Inspection (BSMI) — Taiwan

Please read the appropriate section that corresponds to the marking on your Sun product before attempting to install the product.

FCC Class A Notice

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions:

1. This device may not cause harmful interference.
2. This device must accept any interference received, including interference that may cause undesired operation.

Note: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy, and if it is not installed and used in accordance with the instruction manual, it may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user will be required to correct the interference at his own expense.

Modifications: Any modifications made to this device that are not approved by Sun Microsystems, Inc. may void the authority granted to the user by the FCC to operate this equipment.

ICES-003 Class A Notice - Avis NMB-003, Classe A

This Class A digital apparatus complies with Canadian ICES-003.

Cet appareil numérique de la classe A est conforme à la norme NMB-003 du Canada.

VCCI基準について

クラスA VCCI基準について

クラスA VCCIの表示があるワークステーションおよびオプション製品は、クラスA情報技術装置です。これらの製品には、下記の項目が該当します。

この装置は、情報処理装置等電波障害自主規制協議会(VCCI)の基準に基づくクラスA情報技術装置です。この装置を家庭環境で使用すると電波妨害を引き起こすことがあります。この場合には使用者が適切な対策を講ずるよう要求されることがあります。

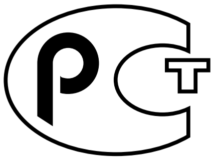
BSMI Class A Notice

The following statement is applicable to products shipped to Taiwan and marked as Class A on the product compliance label.

警告使用者：

這是甲類的資訊產品，在居住的環境中使用時，可能會造成射頻 干擾，在這種情況下，使用者 會被要求採取某些適當的對策。

GOST-R Certification Mark



Contents

Preface

This manual is intended for users who will perform standard configuration and implementation tasks with the Sun^{TM} Secure Application Switch.

The Sun Secure Application Switch is an intelligent application switch that provides advanced Layer 3 to Layer 7 (L3 to L7) load balancing and advanced Secure Sockets Layer (SSL) acceleration with reencryption. The switch provides these services on a flexible, virtualized basis, within the convenience of a single enclosure, and with industry-leading speed, security, and availability.

The Sun Secure Application Switch includes the N1000 Series and the N2000 Series. The N1000 Series includes the N1400 switch and the N1216 switch. The N2000 Series includes the N2040 switch and the N2120 switch. When it is necessary to differentiate between the three switches, the model numbers are used in this manual.

This manual covers the configuration tasks that you will usually perform after you install the hardware and run the setup script. See the *Sun Secure Application Switch – Getting Started Guide* for information about hardware installation and the setup script.

How This Document Is Organized

This manual includes the following topics:

- Chapter 1 describes how to configure vSwitches and vRouters.
- Chapter 2 describes how to configure IP Interfaces.
- Chapter 3 describes the types of load balancing and provides configuration steps.
- Chapter 4 describes persistence.
- Chapter 5 describes redundancy.

- Chapter 6 describes server health checks.
- Chapter 7 describes firewall load balance configuration.
- Chapter 8 describes secure sockets layer (SSL) certificates for configuration synchronization.
- Chapter 9 describes configuration synchronization.

Product Web Page

You can access product information, updated documentation, and other relevant information about the Sun Secure Application Switch at:

`http://www.sun.com/products/networking/switches/`

Typographical Conventions

This manual uses the following typographical conventions.

TABLE P-1 Typographical Conventions

Convention	Function	Example
Ctrl+*x*	Indicates a Control key combination	Press Ctrl+C
[*key name*]	Identifies the name of a key to press	Type **`xyz`**, then press [Enter]
brackets [ ]	Indicates an optional argument	`show telnetd sessions [clientIp` *`ipaddress`*`]`
quotes ““	Encloses a field value that contains spaces	`host h1 description “finance server”`

TABLE P-1 Typographical Conventions *(Continued)*

Convention	Function	Example
braces { }	Indicates a required argument with a choice of values; choose one	`ckm import paste pairHalf {privateKey	certificate}`
	Encloses a field value that contains quotations	`objectRule rule1 predicate {URI_QUERY matches "information*"}`
vertical bar |	Separates parameter values. Means "or"	`format {pem	der	iis4	pkcs12	sun}`
Monospaced regular	Screen output, argument keywords, and defined argument values	`switchServices telnetd adminState enabled`
Monospaced italic	Variable; generic text for which you supply a value	`ntpserver id` *`number`*
Monospaced **bold**	User input	`sun>` **`show vSwitch`**

CLI Commands

Command-line interface (CLI) commands are not case sensitive. For example, SWITCHSERVICES is the same as switchServices. However, the text strings that you enter for argument values *are* case sensitive. For example, ENGR and engr represent two different values.

Related Documentation

The Sun Secure Application Switch documentation listed here is available online at the following URL:

`http://www.sun.com/products/networking/switches/`

TABLE P-2 Related Documentation

Title	Part Number	Format	Location[*]
Sun Secure Application Switch – Configuration and Implementation Guide (This document)	819-7595	PDF	Online
Sun Secure Application Switch – Getting Started Guide	819-3042	Printed PDF	Ship Kit Online
Sun Secure Application Switch for V4.0 – Release Notes	819-7244	Printed PDF	Ship Kit Online
Sun Secure Application Switch – Command Reference for V4.0	819-7594	HTML	Online
Sun Secure Application Switch – Online Help for v4.0	819-7596	HTML	Within application

[*] You can also order at no cost a Documentation CD (part number X3797A) that includes these documents, as well as updated MIBs. Go to http://www.sun.com/products/networking/switches for additional information.

How to Obtain Updates From Sun

You can obtain updates and patches from your Sun authorized sales representative, service provider, or by downloading them from the SunSolve Online^{SM} Web site at the following URL:

`http://sunsolve.sun.com/`

For patch information instructions, see the README file that accompanies each patch.

For downloads of released software, visit the Sun Download Center at the following URL:

`http://www.sun.com/downloads`

Contacting Sun Technical Support

If you have technical questions about this product that are not answered in this document, go to

`http://www.sun.com/service/contacting`

Third-Party Web sites

Sun is not responsible for the availability of third-party Web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused by or in connection with the use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. You can submit your comments by going to:

`http://www.sun.com/hwdocs/feedback`

Please include the title and part number of your document with your feedback:

Sun Secure Application Switch – Configuration and Implementation Guide ,
part number 819-7595.

CHAPTER 1

How to Configure the vSwitches and vRouters

This chapter covers the steps that are required to configure the vSwitches and vRouters. Your specific vSwitch and vRouter configuration will depend on your network requirements. For example, your configuration might require three additional shared vRouters to support the operator-defined vSwitches that you configure. Or, your switch configuration might require only the default shared vRouter, which can support multiple operator-defined vSwitches.

For additional information about configuring vSwitches and vRouters and the specific commands cited in this chapter, see the *Sun Secure Application Switch – Command Reference for V4.0*.

Note – Examples throughout this chapter show how to perform configuration tasks using the command-line interface (CLI). For information about using the other methods available, see the *Sun Secure Application Switch – Getting Started Guide*.

About the vSwitches and vRouters

A virtual switch (vSwitch) is a logical partition of the Sun Secure Application Switch physical platform, which allows the switch to operate as if it were a number of independent switches. Creating multiple vSwitches allows you to partition the data center among multiple users and organizations based on the network services and applications they are running.

The Sun Secure Application Switch consists of two types of vSwitches:

- The system vSwitch is a predefined vSwitch that provides the management, switching, and routing interfaces to the Internet or shared networks. The system vSwitch provides the management vRouter and one preconfigured shared vRouter.
- An operator-defined vSwitch supports one vRouter, which provides connections to the servers in the backend networks.

To configure the vSwitches and vRouters, perform the following tasks:

1. **Set up the system vSwitch.**

   To set up the system vSwitch, configure the management vRouter and the shared vRouter. See “Setting Up the System vSwitch” on page 2.

2. **Create the operator-defined vSwitches.**

   To create the operator-defined vSwitches, configure the default vRouter and allocate port and memory resources. See “Creating Operator-Defined vSwitches” on page 4.

Setting Up the System vSwitch

To set up the system vSwitch, perform the following steps:

1. **Configure the management vRouter.**

   If you have not assigned an IP address to the management vRouter through the setup script, use the following CLI session as a guide to assign an IP address. The session uses the default (ethMgmt.1) 10/100-Mbps Ethernet management port.

```
sun> enable
sun# config
sun(config)# vSwitch system
sun(config vSwitch-system)# vRouter management
sun(config vSwitch-system vRouter management)# ip
sun(config-vSwitch-system vRouter management ip)# address ethMgmt.1
10.10.10.15 255.255.255.0
```

2. **Set up a default router to the local gateway.**

   A default router allows access to the management port from management stations on different subnets. Use the following CLI session as a guide for setting up a default router.

```
sun(config-vSwitch-system vRouter management)# ip
sun(config-vSwitch-system vRouter management ip)# route static
destAddr 0.0.0.0 mask 0.0.0.0 nextHop 10.10.10.10
```

3. **Configure the shared vRouters.**

   By default, one shared vRouter is automatically created and called shared. The switch supports up to three additional shared vRouters.

   Use the following CLI session as a guide for configuring the default vRouter named `shared` on the system vSwitch. The shared vRouter routes traffic over interface `eth.1.4`.

```
sun(config)# vSwitch system
sun(config-vSwitch-system)# vRouter shared
sun(config-vSwitch-system vRouter-shared)# ip interface eth.1.4
sun(config-vSwitch-system vRouter-shared)# ip address eth.1.4
10.10.10.12 255.255.255.0
sun(config-vSwitch-system vRouter-shared) exit
```

   You can also create up to three additional shared vRouters in addition to the default. If you configure multiple shared vRouters, each additional shared vRouter must have a unique name.

   The following CLI session is an example of configuring an additional vRouter named `backend` on the system vSwitch. In this example, the vRouter `backend` routes traffic over interface `eth.1.16`.

```
sun(config-vSwitch-system) vRouter backend
sun(config-vRouter-backend)# ip interface eth.1.16
sun(config-vRouter-backend)# ip address eth.1.16 10.10.10.1
255.255.255.0
sun(config-vSwitch-system vRouter-backend) exit
```

Creating Operator-Defined vSwitches

Creating multiple operator-defined vSwitches allows you to partition the data center among different customers based on the network services and the applications they are running.

1. **To create an operator defined vSwitch and to configure the default vRouter, perform the following steps:**

   Use the following CLI session, which creates a vSwitch called `e-commerce`, as a guide for creating an operator-defined vSwitch.

   ```
   sun(config)# vSwitch e-commerce
   sun(config-vSwitch-e-commerce)# exit
   ```

   Use the `show vswitch` command to display the vSwitch attributes.

   ```
   sun(config)# show vswitch
   Name:               e-commerce
   ID:                 2
   Description:        e-commerce
   Admin State:        enabled
   Operational Status: up
   ```

   After you have created the operator-defined vSwitch, continue with the following steps.

2. **Configure the default vRouter.**

   Each operator-defined vSwitch supports a single default vRouter. The default vRouter provides connections to the servers in the backend networks. On a default vRouter, you configure the Ethernet or VLAN interfaces, their associated IP addresses, and IP routing protocols.

CHAPTER **2**

How to Configure IP Interfaces

This chapter covers the steps that are required to configure the IP interfaces. Based on your network configuration, you might need to configure IP interfaces and assign IP addresses on the vRouter. It is over these IP interfaces that inbound and outbound traffic can be filtered using access control lists (ACLs).

For additional information about configuring IP interfaces and the specific commands cited in this chapter, see the *Sun Secure Application Switch – Command Reference for V4.0*.

Note – Examples throughout this chapter show how to perform configuration tasks using the command-line interface (CLI). For information about using the other methods available, see the *Sun Secure Application Switch – Getting Started Guide*.

About IP Interfaces

The IP routing function in the Sun Secure Application Switch supports three types of interfaces:

- 10/100-Mbps or Gigabit Ethernet ports - see “Configuring IP Over Ethernet Interfaces” on page 6.
- Virtual LANs (VLANs) - see “Configuring IP Over VLAN Over Ethernet Interfaces” on page 7.
- Link aggregation groups (LAGs) - see “Configuring IP Over VLAN Over LAGs” on page 8.

You can configure up to 128 IP interfaces per vRouter.

To assign an IP address (and optionally subnet mask) to an interface, you must first configure the interface. Typically, you configure the actual IP interfaces on the default vRouter; however, your specific network design determines whether you configure IP interfaces on the shared vRouters or on the default vRouter.

The following sections show how to configure the three types of IP interfaces.

Configuring IP Over Ethernet Interfaces

Ethernet ports are numbered according to the system slot and their position in the slot. Since the switch is a single-slot device, all Ethernet ports are in slot 1.

Interfaces such as eth.1.4 are Ethernet interfaces to customer networking equipment (switches, routers, or hosts) in the data center. On the switch, the interface labeled eth.1.4 represents the fourth physical Ethernet interface on the switch.

The following CLI configuration session adds three IP interfaces to the default vRouter directly over an Ethernet interface and assigns IP addresses and network masks.

```
sun(config-vSwitch-e-commerce-vRouter-default) ip
...vRouter-default ip)# interface eth.1.1
...vRouter-default ip)# address eth.1.1 10.10.20.1 255.255.255.0

...vRouter-default ip)# interface eth.1.2
...vRouter-default ip)# address eth.1.2 10.10.30.1 255.255.255.0

...vRouter-default ip)# interface eth.1.3
...vRouter-default ip)# address eth.1.3 10.10.40.1 255.255.255.0
```

Use the `show interface` and `show address` commands to verify the interfaces and addresses.

```
      sun(config-vSwitch-e-commerce vRouter-default ip)# show interface

If Name      Admin State      Oper Status     MTU      Phys Addr
-------      -----------      -----------     ----     ----------
eth.1.1      enabled          up              1500     00:07:82:00:03:cl
eth.1.2      enabled          up              1500     00:07:82:00:03:c2
eth.1.3      enabled          up              1500     00:07:82:00:03:c3

      sun(config-vSwitch-e-commerce vRouter-default ip)# show address

if Index      IP Address      Subnet Mask     VSRP Redirect
--------      -----------     -----------     --------------
eth.1.1       10.10.20.1      255.255.255.0   disabled
eth.1.2       10.10.30.1      255.255.255.0   disabled
eth.1.3       10.10.40.1      255.255.255.0   disabled
```

For more information about configuring IP interfaces over Ethernet interfaces and assigning IP addresses, see the *Sun Secure Application Switch – Command Reference for V4.0.*

Configuring IP Over VLAN Over Ethernet Interfaces

A virtual LAN (VLAN) is a logical grouping of systems that is not constrained by geographical boundaries. These groupings create a broadcast domain, and function just like a traditional LAN. Systems within the LAN are not necessarily physically co-located, but do not require a router to connect them. Routers are used to connect two separate VLANs. VLANs are interconnected using system bridging software. The Sun Secure Application Switch supports up to 128 VLANs.

VLANs are configured within a virtual router. You can configure a VLAN interface directly on an Ethernet port or on a link aggregation group (LAG). Each VLAN is identified by a VLAN name, and each name must be unique within the physical switch. Each VLAN also has an ID that must be unique.

The following configuration session adds `vlan sales` to the default router and creates the VLAN with two physical interfaces, eth.1.1 and eth.1.2 and assigns an IP address and network mask to the VLAN.

Note – If you are connecting directly to web servers that are not running VLAN tagging, you need to disable tagging on the Ethernet interfaces for the VLAN. See the *Sun Secure Application Switch – Command Reference for V4.0* for more information.

```
    sun(config-vSwitch-e-commerce vRouter-default)# vlan sales 10
    sun(config-vSwitch-e-commerce vRouter-default)# vlan sales
    ...vRouter-default vlan-sales)# interface eth.1.1
    ...vRouter-default vlan-sales)# interface eth.1.2
    ...vRouter-default vlan-sales)# show interface

Vlan Name   If Name    Admin State   Oper Status Tagging
---------   ------     ------------   ---------- --------
sales      eth.1.1    enabled        up        disabled
sales      eth.1.2    enabled        up        disabled

    sun(config-vSwitch-e-commerce vRouter-default vlan-10)# exit
    ...vRouter-default) ip
    ...vRouter-default ip)# interface vlan.sales
    ...vRouter-default ip)# address vlan.sales 10.10.50.1 255.255.255.0
```

For more information about configuring VLAN interfaces over Ethernet interfaces and assigning IP addresses, see the *Sun Secure Application Switch – Command Reference for V4.0.*

Configuring IP Over VLAN Over LAGs

A link aggregation group (LAG) combines multiple Ethernet ports into a virtual link with aggregated bandwidth. The switch treats the set of ports in a LAG as a single port. A LAG can connect to one or more VLANs.

The following configuration session shows how to create a LAG (`lag mktg`) with two physical interfaces, eth.1.3 and eth.1.4, connect the LAG to VLAN sales, and assign an IP address and network mask to the VLAN.

```
sun(config)# lag mktg
sun(config-lag-mktg)# interface eth.1.3
sun(config-lag-mktg)# interface eth.1.4
sun(config-lag-mktg)# exit

sun(config)# vSwitch mktg
sun(config-vSwitch-mktg)# vRouter default
sun(config-vSwitch-sales vRouter-default)# vlan sales
...vRouter-default vlan-sales)# interface lag.mktg
```

```
...vRouter-default vlan-sales)# exit
...vRouter-default) ip
...vRouter-default ip)# interface vlan.sales
...vRouter-default ip)# address vlan.sales 10.10.60.1 255.255.255.0
```

For more information about configuring VLAN over LAG interfaces and assigning IP addresses, see the *Sun Secure Application Switch – Command Reference for V4.0.*

CHAPTER 3

How to Configure Load Balancing

This chapter covers the steps that are required to implement load balancing. Load balancing allows you to configure the application switch to balance session traffic among and between a pool of servers providing services. Load balancing improves network performance by distributing traffic so that individual servers are not overwhelmed with network traffic. The switch directs user session traffic to an appropriate server, based on a variety of load-balancing algorithms. Load balancing allows for increased efficiency of server utilization and network bandwidth, increases reliability with fewer service disruptions, and improves server scalability.

For additional information about configuration tasks, and the specific commands cited in this chapter, see the *Sun Secure Application Switch – Command Reference for V4.0*.

Note – Examples throughout this chapter show how to perform configuration tasks using both the command-line interface (CLI) and the Web interface. For information about using the other methods available, see the *Sun Secure Application Switch – Getting Started Guide*.

About Load Balancing

The Sun Secure Application Switch allows you to load balance Layer 3 through Layer 7 traffic over the operator-defined vSwitches and the web hosts connected to them. The Sun Secure Application Switch Server Load Balancing application distributes TCP/UDP packets destined toward a single virtual IP (VIP) address across a group of real services, each defined by a real IP address. When the switch receives a packet addressed to a virtual service configured for load balancing, the switch redirects the packet to the appropriate real service and performs network address translation (NAT) on the IP address before returning the packet to the client.

This section describes a basic load-balancing configuration based on a destination IP address and a weighted hash algorithm and documents the configuration steps.

To configure load balancing on the Sun Secure Application Switch, perform the following tasks:

1. **Add hosts (web and applications servers) to the vSwitch.**

   A host typically resides on the backend network and is identified by a host name and an IP address. These IP addresses are kept private and are not exposed to the Internet. See "How to Add a Host to the vSwitch" on page 13.

2. **Define the real services that are running on these hosts.**

   A real service, associated with a host, defines the expected type of inbound and outbound traffic processed by the host, as identified by the application port. Real services have assigned weights when they participate in load-balancing groups. See "How to Define a Real Service" on page 14.

3. **Create service groups for fulfilling web service requests.**

   A service group combines one or more real service definitions into a group. A service group assigns a particular load-balancing algorithm to the services in the group, along with other configurable characteristics. See "How to Create a Service Group" on page 15.

4. **Configure virtual services.**

   A virtual service contains the client visible configuration attributes for the load balancer. A virtual service defines the virtual IP address (VIP) and specifies the application service type (L3SLB, L4SLB, L4SLB_ADV, L4SLB_SSL, HTTP, HTTPS, TDLB, FTPLB, and RTSPLB). A virtual service also defines the vRouter for client traffic. See "How to Create a Virtual Service" on page 18.

The following sections show the CLI sessions required to implement load balancing.

How to Add a Host to the vSwitch

The host is a machine such as a backend server with an assigned IP address. Each host requires a name and an IP address. The IP address defined is the IP address of the actual server to be load-balanced.

Each host definition requires the following elements:

- Name
- IP address

The host definition includes a set of default settings that you can customize, including:

- A description of the host.
- Administrative setting (enabled, disabled).
- The vRouter on which the server will be receiving traffic. The default is the vRouter of the current vSwitch.

Refer to the *Sun Secure Application Switch – Command Reference for V4.0* or the *Sun Secure Application Switch – Online Help for V4.0* for detailed descriptions of the parameters available for the `host` command.

Configuration steps

To configure a host, perform the following steps:

1. **Select Host from the Load Balance menu in the web interface or use the** `Host` **command in the CLI.**

2. **Enter a host name and the IP address of the host.**

3. **Change the default settings of the other optional parameters, if appropriate.**

CLI Session

The following CLI session adds the host servers `host_1`, `host_2`, and `host_3` to the network and assigns an IP address to each host.

```
sun(config-vSwitch-e-commerce)# loadbalance
sun(config vSwitch-e-commerce loadBalance)# host host_1 10.10.50.2
sun(config vSwitch-e-commerce loadBalance)# host host_2 10.10.50.3
```

```
              sun(config vSwitch-e-commerce loadBalance)# host host_3 10.10.50.4
              sun(config-vSwitch-e-commerce loadBalance)# show host

Name      IP Address          Admin State    Description    vRouter
-------   -----------         -----------    -----------    -------
host_1    10.10.50.2          enabled        N/A            e-commerce:default
host_2    10.10.50.3          enabled        N/A            e-commerce:default
host_3    10.10.50.4          enabled        N/A            e-commerce:default
```

How to Define a Real Service

Real service definitions are associated with each host that you are configuring on the network. Real service definitions (within a service group) provide some of the required input parameters for the load-balancing algorithm that you choose.

Each real service definition requires the following elements:

- Name
- Host name

Note – Each real service requires a host definition. However, multiple real service definitions can be defined on the same host.

- Protocol (defaults to TCP)
- Port (defaults to 80) - the port does not need to match the virtual service destination
- Weight (defaults to 1) - the higher number accepting more load-balancing requests compared to other real service definitions with lower weights, or a dynamic weight setting derived from the server health checks

Refer to the *Sun Secure Application Switch – Command Reference for V4.0* or the *Sun Secure Application Switch – Online Help for V4.0* for detailed descriptions of all parameters that are available for the `realService` command.

Configuration Steps

To create a real service, perform the following steps:

1. **Select Real Service from the Load Balance menu in the web interface or use the** `realService` **command in the CLI.**
2. **Enter a** `realService` **name, a host name, a protocol, a port, and a weight.**

3. **Change the default settings of the other optional parameters, as appropriate.**

CLI Session

The following CLI session creates the real service named `rs1` on the host named `host_1`. In this example, `host_1` balances TCP traffic received on TCP port 80 using a load-balancing weight of 1. Use the `show` command to display the real service definitions for this host and others that you create.

```
sun(config-vSwitch-e-commerce loadBalance)# realService rs1 host_1
protocol TCP port 80 weight 1
```

```
sun(config-vSwitch-e-commerce loadBalance)# show realService
```

```
Name:                                 rs1
Host Name:                            host_1
Protocol:                             TCP
Port:                                 80
Weight In ServiceGroup:               1
Admin State:                          enabled
Disable Delay:                        0
inLine SHC Failure Rate Threshold:    1
Client Address Translation Mask:      disabled
Bridge Mode:                          disabled
Encryption:                           unencrypted
Virtual Services:
Oper Status:                          inactive
Health Check Port:                    80
```

How to Create a Service Group

A service group places real service definitions into one common grouping, and specifies the load-balancing algorithm to be used in making traffic-balancing decisions across the hosts. Optionally, you can configure standby real services that the system brings into service if one of the configured real services fails.

Each service group definition requires the following elements:

- Name
- Load-balance type (such as roundRobin, weightedRandom, weightedHash, leastConnections, and weightedLeastConnections)

Note – The load-balance type is the algorithm used for load-balancing decisions. See "Load-Balancing Algorithms" on page 17.

- Real services (list of real services to load balance across)

Each service group requires a name, a selected load-balancing algorithm, and the list of realServices that participate in the group. The service group definition includes a set of default settings that you can customize, for example:

- Administrative settings (current enabled/disabled state)
- Load-balancing metrics
- Server health profile name
- Customized SSL settings for this service group (regeneration, authentication, SSL protocols, ciphers, renegotiation)
- Standby real services

Refer to the *Sun Secure Application Switch – Command Reference for V4.0* or the *Sun Secure Application Switch – Online Help for V4.0* for detailed descriptions of all parameters that are available for the `serviceGroup` command.

Configuration Steps

To configure a service group, perform the following steps:

1. **Select Service Group from the Load Balance menu in the web interface or use the** `serviceGroup` **command in the CLI.**

2. **Configure a serviceGroup name, load-balance type, and list of real services.**

3. **Change the default settings of the other optional parameters, if appropriate.**

CLI Session

The following CLI session creates the service group named `imageServers`, specifies weighted hash load balancing, and adds the real services that will load balance traffic.

```
sun(config-vSwitch-e-commerce loadBalance)# serviceGroup
imageServers weightedHash {rs1 rs2 rs3}
```

```
sun(config-vSwitch-e-commerce loadBalance)# show serviceGroup
```

```
Name:                     imageServers
Load Balance Type:        weightedHash
Configured Real Services: rs1; rs2; rs3
```

```
Active Real Services:      N/A
Active RS Activation:      asNeeded
Admin State:               enabled
Virtual Services:
Health Check Name:
In-Line Health Check:      enabled
Failover Retry Count:      1
Flash Crowd Threshold:     1
Oper Status:               active
```

Load-Balancing Algorithms

Load-balancing algorithms are specific to the application service type of the virtual service that the service group is in.

There are five load-balancing algorithms that you can specify in the service group:

- **Round-robin –** The round-robin algorithm distributes traffic sequentially to the next server in the service group. All servers are treated equally, regardless of the number of inbound connections or the response time.
- **Weighted random –** The weighted random algorithm distributes traffic to web servers randomly using weight settings.
- **Weighted hash** – The weighted hash algorithm attempts to distribute traffic evenly across a service group.
- **Least connections –** The least connections algorithm dynamically directs traffic to the server with the least number of active connections.
- **Weighted least connections –** The weighted least connections algorithm distributes traffic to a server with the least amount of current active connections relative to its importance in the server group.

Weighted Algorithms

Note – For each weighted algorithm, you can assign static weights or a dynamic weight.

When configuring a real service with a static load-balancing weight (instead of using dynamic weight), you should consider that server's ability to handle more or less traffic than other servers in the group. If a server is capable of handling more traffic, then set the server weight to a higher weight (by number) than those weights assigned to other servers in the group. Hashing of the client IP address is used so that generally sessions from a given client will be balanced to the same server.

The system uses a lowest latency algorithm in weight calculations if the weight is set to dynamic using the `realService` command. (This setting is ignored if a static value is configured for the weight.) Lowest latency computes the response time to and from a server, and uses that value to determine the current fastest real service.

How to Create a Virtual Service

A virtual service contains the client visible configuration attributes for the server load balancer. The virtual service provides the virtual IP address (VIP) to the load balancer and specifies the application service type for the service group. A virtual service also defines the vRouter on which the client traffic will be received.

Each virtual service definition requires the following elements:

- Virtual service name
- Service type

Refer to the *Sun Secure Application Switch – Command Reference for V4.0* or the *Sun Secure Application Switch – Online Help for V4.0* for detailed descriptions of all parameters that are available for the `virtualService` command.

Configuration Steps

To configure a virtual service, perform the following steps:

1. **Select Virtual Service from the Load Balance menu in the web interface or use the** `virtualService` **command in the CLI.**
2. **Configure a virtual service name and service type.**

   The service type you select determines the parameters available for the virtual service.
3. **Change the default settings of the other optional parameters, if appropriate.**

CLI Session

The following CLI session creates the virtual service named `e-commerceNet`, specifies service type `L4SLB`, sets the VIP to `10.10.50.11,` and defines the service group `imageServers`.

By default, the switch links the operator-defined e-commerce vSwitch to the system:shared vRouter on the system vSwitch.

```
sun(config-vSwitch-e-commerce loadBalance)# virtualService
e-commerceNet L4SLB 10.10.50.11 imageServers
```

The show virtualService command displays these settings and the default virtual service settings.

```
sun(config-vSwitch-e-commerce loadBalance)# show virtualService
```

```
Name:                             e-commerceNet
Service Type:                     L4SLB
IP Address:                       10.10.50.11
Protocol:                         TCP
Port:                             80
Service Group:                    imageServers
Admin State:                      enabled
Disable Delay:                    0
vRouter:                          system:shared
Client Source IP Address Range: 0.0.0.0-255.255.255.255
SYN Rate Limit:                   unlimited
Client Address Hash Mask:         0.0.0.0-255.255.255.255
Client Port Hash:                 enabled
Oper Status:                      active
Oper Message:                     Operational; Server Health Checking
                                  not configured on all serviceGroups
Total Real Services:              1
Active Real Services:             1
```

How to View Virtual Service Status

To verify the virtual service works properly, the Oper Status and Oper Message fields from the virtual service output includes information to diagnose configuration issues.

In the Oper Status field, the status will be either active or inactive. If the status is active, this means the virtual service is working and is also capable of passing traffic.

If the status is active, the message in the Oper Message field will be either:

- Operational
- Operational; Server Health Checking not configured on all serviceGroups

If the status in the Oper Status field is inactive, the Oper Message field will display diagnostic information about the Oper Status field.

CHAPTER 4

Using Persistence

In many authenticated Web-based applications, it is necessary to provide a persistent connection between a client and the connected content server. Because HTTP does not carry any state information for these applications, the browser must be mapped to the same server for each HTTP request until the transaction is complete. This mapping ensures that the client traffic is not load balanced mid-session to a different server, forcing you to restart the entire transaction. This chapter describes how the Sun Secure Application Switch implements persistence.

Note – Examples throughout this chapter show how to perform configuration tasks using the command-line interface (CLI). For information about using the other methods available, see the *Sun Secure Application Switch – Getting Started Guide.*

About Persistence

Persistence enables you to configure the switch to redirect requests from a client to the same real service that initially handled the request. Persistence is an important consideration for administrators of e-commerce web sites, where a server might have data associated with a specific user that is not dynamically shared with other servers at the site. The following sections describe how the Sun Secure Application Switch uses cookies to implement session persistence.

Cookie Persistence Overview

When configured, the Sun Secure Application Switch load balancer inserts a cookie into HTTP server response packets returned to the client. This cookie, (called a “switch-managed cookie”) identifies the original server to the server load-balancing

(SLB) application. The client stores this cookie and includes it in subsequent HTTP client requests to the same server. When the load balancer receives a request containing this cookie, the load balancer deciphers the cookie, and then forwards the traffic to the indicated web server.

During the session, HTTP requests use the same cookie, keeping the connection persistent until the client closes the session.

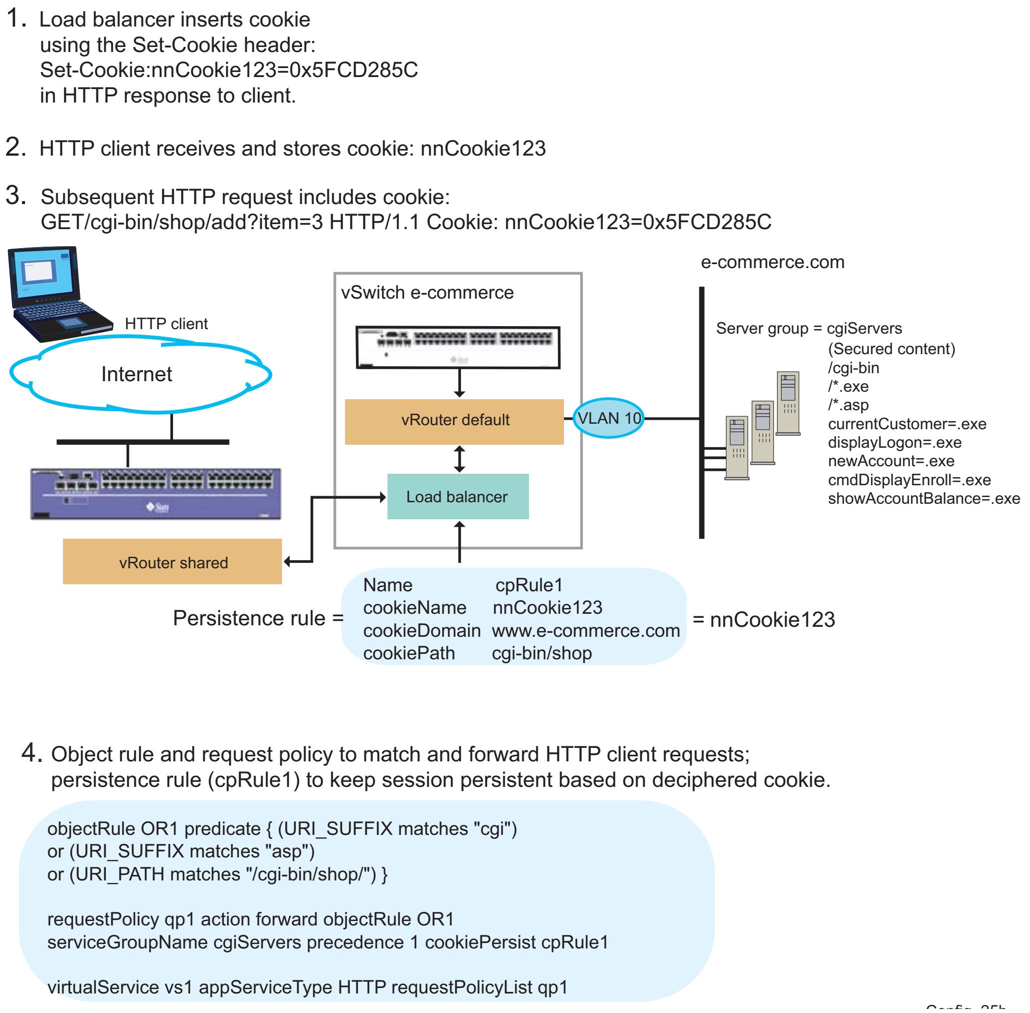
The following illustration shows a sample network using a `cookiePersistence` command and a sample object rule configuration session. In this example the following occurs:

1. The load balancer uses the cookie persistence rule (cp1Rule1) to define the cookie to be inserted into the HTTP response.

2. The HTTP client receives and stores the cookie.

3. The next HTTP request from the client includes the cookie as a separate header.

4. The load balancer uses the cookie persistence value to forward the request to the destination server.



1. Load balancer inserts cookie
   using the Set-Cookie header:
   Set-Cookie:nnCookie123=0x5FCD285C
   in HTTP response to client.

2. HTTP client receives and stores cookie: nnCookie123

3. Subsequent HTTP request includes cookie:
   GET/cgi-bin/shop/add?item=3 HTTP/1.1 Cookie: nnCookie123=0x5FCD285C

Name	cpRule1
cookieName	nnCookie123
cookieDomain	www.e-commerce.com
cookiePath	cgi-bin/shop

4. Object rule and request policy to match and forward HTTP client requests;
   persistence rule (cpRule1) to keep session persistent based on deciphered cookie.

```
objectRule OR1 predicate { (URI_SUFFIX matches "cgi")
or (URI_SUFFIX matches "asp")
or (URI_PATH matches "/cgi-bin/shop/") }

requestPolicy qp1 action forward objectRule OR1
serviceGroupName cgiServers precedence 1 cookiePersist cpRule1

virtualService vs1 appServiceType HTTP requestPolicyList qp1
```

CHAPTER **5**

How to Configure Redundancy

Configuring redundancy in your network can help you to ensure high availability by eliminating or minimizing a single point of failure in your network. To implement redundancy, you configure the redundancy protocols of the Sun Secure Application Switch. This chapter covers the configuration tasks that you will perform to implement redundancy in your network.

For additional information about configuration tasks, and the specific commands cited in this chapter, see the *Sun Secure Application Switch – Command Reference for v4.0*.

Note – Examples throughout this chapter show how to perform configuration tasks using the command-line interface (CLI). For information about using the other methods available, see the *Sun Secure Application Switch – Getting Started Guide*.

About Redundancy

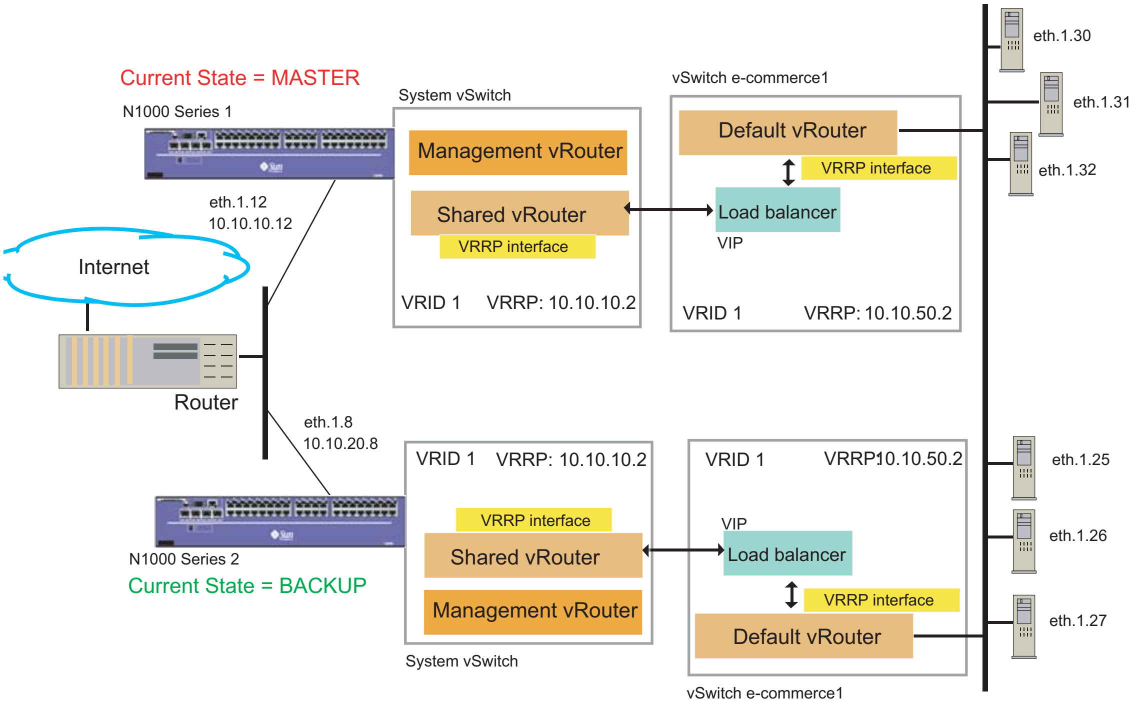
Configuring the Virtual Service Redundancy Protocol (VSRP) and the Virtual Router Redundancy Protocol (VRRP) on two Sun Secure Application Switches provides link-level and service-level failover capabilities between the two switches.

- Using two application switches, VRRP provides continued service to a redundant switch when a link goes down. When router interfaces fail, VRRP provides access to an active default gateway. For more information, see “Configuring VRRP” on page 26.
- VSRP guarantees continued operation for the virtual services during a switch failure. The method VSRP uses to ensure continued operation depends on the mode of operation (All or None Mode or Per VSwitch Mode) you select when you configure VSRP. For more information about the modes of operation and how VSRP ensures continued operation, see “Configuring VSRP” on page 31.

Configuring VRRP

Note – VRRP must be configured on the frontend and backend networks for VSRP to provide service-level failover. Running VRRP on the frontend network ensures that a vRouter will always respond to Address Resolution Protocol (ARP) requests for VIPs.

The following figure illustrates a network where the backend servers use a single virtual LAN (VLAN 10) and the servers are connected to L2 switching equipment. This section provides example CLI sessions showing how to configure VRRP in the two physical switches (one is called the “N1000 Series 1” and one is called the “N1000 Series 2”) shown in the sample network in the figure. See “CLI Sessions for the N1000 Series 1 Switch” on page 27 and “CLI Sessions for the N1000 Series 2 Switch” on page 29 for examples of how to configure VRRP for each physical switch.



CLI Sessions for the N1000 Series 1 Switch

Steps 1 through 3 show the CLI sessions for configuring VRRP on the system vSwitch/shared vRouter. Steps 4 through 6 show the CLI sessions for configuring VRRP on the operator-defined e-commerce vSwitch default vRouter.

1. **Configure the physical Sun Secure Application Switch interfaces and IP addresses.**

   The following session configures the IP interface to the Internet on the system vSwitch/shared vRouter.

```
sun(config)# vSwitch system
sun(config-vSwitch-system)# vRouter shared
...vRouter-shared)# ip interface eth.1.12
...vRouter-shared)# ip address eth.1.12 10.10.10.12 255.255.255.0
...vRouter-shared)# show ip address

 If Index     IP Address       Subnet Mask    Redirect Traffic
 --------    -----------       -----------    ----------------
 eth.1.12     10.10.10.12      255.255.255.0  disabled
```

2. **Configure the VRRP interface, VRRP ID, one or more VRRP IP addresses, and optional settings.**

   The following CLI session configures the VRRP interface on the shared vRouter. Configuring VRRP on the shared and default vRouter interface creates the VRRP virtual router. See Step 5 for the default vRouter configuration session.

   Note – The VRID and VRRP IP address on this vRouter must match the VRID and VRRP IP address on the redundant vRouter.

```
sun(config-vSwitch-system vRouter-shared)# vrrp
sun(config-vSwitch-system vRouter-shared vrrp)# interface eth.1.12
vrid 1 10.10.10.2 adminState enabled priority 100 interval 5 preempt
true
sun(config-vSwitch-system vRouter-shared vrrp)# show interface

 If Index:            eth.1.12
 VRID:                1
 IP Addresses:        10.10.10.2
 Primary IP Address:  10.10.10.12
 Admin State:         enabled
 Oper State:          master
 Priority:            100
 Actual Priority:     100
 Interval:            5
```

```
Preempt Mode:        true
MAC Address:         00:00:5e:00:01:01
IP Address Count:    3
Master's IP Address: 10.10.10.12
Up Time:             3/10/2003-11:45:11
```

3. **(Optional) Set the VSRP preference and enable trap generation.**

   When VSRP is configured for All or None Mode, the VSRP preference indicates the incremental increase in VRRP preference that this switch advertises when it is the VSRP master. This increases the likelihood that the master VSRP switch will also be the VRRP master.

```
sun(config-vSwitch-system vRouter-shared)# vrrp vsrpPreference 10
sun(config-vSwitch-system vRouter-shared)# vrrp traps enabled
```

Note – Steps 4 through 6 show how to configure the e-commerce vSwitch default vRouter.

---

4. **Configure the VLAN interfaces on the e-commerce1 vSwitch default vRouter.**

```
sun(config-vSwitch-e-commerce1 vRouter-default)# vlan sales 10
sun(config-vSwitch-e-commerce1 vRouter-default)# vlan sales
...vRouter-default vlan-sales)# interface eth.1.30
...vRouter-default vlan-sales)# interface eth.1.31
...vRouter-default vlan-sales)# interface eth.1.32
...vRouter-default vlan-sales)# show interface
```

```
Vlan Name       ifName      Admin State     Oper Status     Tagging
----------      -------     -----------     -----------     --------
sales           eth.1.30    enabled         up              disabled
sales           eth.1.30    enabled         up              disabled
sales           eth.1.32    enabled         up              disabled
```

```
...vRouter-default vlan-10)# exit
...vRouter-default) ip
...vRouter-default ip)# interface vlan.sales
...vRouter-default ip)# address vlan.sales 10.10.50.1 255.255.255.0
```

5. **Configure the VRRP interface, VRRP ID, VRRP IP addresses, and optional settings.**

```
sun(config-vSwitch-e-commerce1 vRouter-default)# vrrp
sun(config-vSwitch-e-commerce1 vRouter-default vrrp)# interface
vlan.sales vrid 1 10.10.50.2 adminState enabled priority 100
interval 5 preempt true
```

```
sun(config-vSwitch-e-commerce1 vRouter-default vrrp)# show interface

If Index:            vlan10
VRID:                1
IP Addresses:        10.10.50.2
Primary IP Address:  10.10.50.1
Admin State:         enabled
Oper State:          master
Priority:            100
Actual Priority:     100
Interval:            5
Preempt Mode:        true
MAC Address:         00:00:5e:00:01:01
IP Address Count:    3
Master's IP Address: 10.10.50.1
Up Time:             3/10/2003-11:45:11
ICMP Echo:           disabled
```

6. **(Optional) Set the VSRP Preference and enable trap generation.**

```
sun(config-vSwitch-e-commerce1 vRouter-default)# vrrp vsrpPreference
10
sun(config-vSwitch-e-commerce1 vRouter-default)# vrrp traps enabled
```

CLI Sessions for the N1000 Series 2 Switch

Steps 1 through 3 show the CLI sessions for configuring VRRP on the system vSwitch shared vRouter on the second physical switch. Steps 4 through 6 show the CLI sessions for configuring VRRP on the e-commerce vSwitch default vRouter.

1. **Configure the physical Sun Secure Application Switch interfaces and IP addresses.**

```
sun(config)# vSwitch system

sun(config-vSwitch-system)# vRouter shared

sun(config-vSwitch-system vRouter-shared)# ip interface eth.1.8

...vRouter-shared)# ip address eth.1.8 10.10.20.8 255.255.255.0

...vRouter-shared)# show ip address


 If Index      IP Address       Subnet Mask       Redirect Traffic
 --------      ----------       -----------       ----------------
 eth.1.8       10.10.20.8       255.255.255.0     enabled
```

2. **Configure the VRRP interface, VRRP ID, VRRP IP addresses, and optional settings.**

```
sun(config-vSwitch-system vRouter-shared)# vrrp
sun(config-vSwitch-system vRouter-shared vrrp)# interface eth.1.8
vrid 1 10.10.10.2 adminState enabled priority 50
interval 5 preempt true
```

```
sun(config-vSwitch-system vRouter-shared vrrp)# show interface


If Index:            eth.1.8
VRID:                1
IP Addresses:        10.10.10.2
Primary IP Address:  10.10.20.8
Admin State:         enabled
Oper State:          backup
Priority:            50
Actual Priority:     50
Interval:            5
Preempt Mode:        true
MAC Address:         00:00:5e:00:01:01
IP Address Count:    3
Master's IP Address: 10.10.10.12
Up Time:             3/10/2003-11:45:11
ICMP Echo:           disabled
```

3. **(Optional) Set the VSRP Preference and enable trap generation.**

```
sun(config-vSwitch-system vRouter-shared)# vrrp vsrpPreference 1
```

```
sun(config-vSwitch-system vRouter-shared)# vrrp traps enabled
```

4. **Configure the VLAN interfaces on the e-commerce1 vSwitch default vRouter.**

```
        sun(config-vSwitch-e-commerce1 vRouter-default)# vlan sales 10
        sun(config-vSwitch-e-commerce1 vRouter-default)# vlan sales
        ...vRouter-default vlan-sales)# interface eth.1.25
        ...vRouter-default vlan-sales)# interface eth.1.26
        ...vRouter-default vlan-sales)# interface eth.1.27
        ...vRouter-default vlan-sales)# show interface

Vlan Name       ifName      Admin State     Oper Status     Tagging
----------      -------     -----------     -----------     --------
sales           eth.1.25    enabled         up              disabled
sales           eth.1.26    enabled         up              disabled
sales           eth.1.27    enabled         up              disabled


      ...vRouter-default vlan-sales)# exit
      ...vRouter-default)# ip
      ...vRouter-default ip)# interface vlan.sales
      ...vRouter-default ip)# address vlan.sales 10.10.50.1 255.255.255.0
```

5. **Configure the VRRP interface, VRRP ID, VRRP IP addresses, and optional settings**.

```
sun(config-vSwitch-e-commerce1 vRouter-default)# vrrp
sun(config-vSwitch-e-commerce1 vRouter-default vrrp)# interface
vlan.sales vrid 1 10.10.50.2 adminState enabled priority 50 interval
5 preempt true
sun(config-vSwitch-e-commerce1 vRouter-default)# show interface
If Index:            vlan.10
VRID:                2
IP Addresses:        10.10.50.2
Primary IP Address:  10.10.50.1
Admin State:         enabled
Oper State:          backup
Priority:            50
Actual Priority:     50
Interval:            5
Preempt Mode:        true
MAC Address:         00:00:5e:00:01:01
IP Address Count:    3
Master's IP Address: 10.10.50.1
Up Time:             3/10/2003-11:45:11
ICMP Echo:           disabled
```

6. **(Optional) Set the VSRPPreference and enable trap generation.**

```
sun(config-vSwitch-e-commerce1 vRouter-default)# vrrp
vsrpPreference 1
sun(config-vSwitch-e-commerce1 vRouter-default)# vrrp traps enabled
```

For detailed information on the optional VRRP configuration settings, refer to the *Sun Secure Application Switch – Command Reference for v4.0.*

Configuring VSRP

Virtual Service Redundancy Protocol (VSRP) is a protocol that provides redundancy for virtual services between two configured VSRP peers. When configuring VSRP, you can choose one of two possible modes of operation via the `electionMode` setting:

- **All or None Mode** – In an All or None Mode configuration of two Sun Secure Application Switches, one physical switch is the active (master) switch and one physical switch is the standby switch. If the master switch is unable to maintain active virtual services, all vSwitches on the affected switch will transition to the alternate (standby) physical switch, which then becomes the master switch. See "Configuring All or None Mode" on page 32.

- **Per vSwitch Mode** – In a Per vSwitch Mode configuration of two Sun Secure Application Switches, both switches can simultaneously host active vSwitches and standby vSwitches. See “Configuring Per vSwitch Mode” on page 34.

When configuring VSRP in either mode, you must configure VRRP on vRouters that are used by virtual services and real services in the switch.

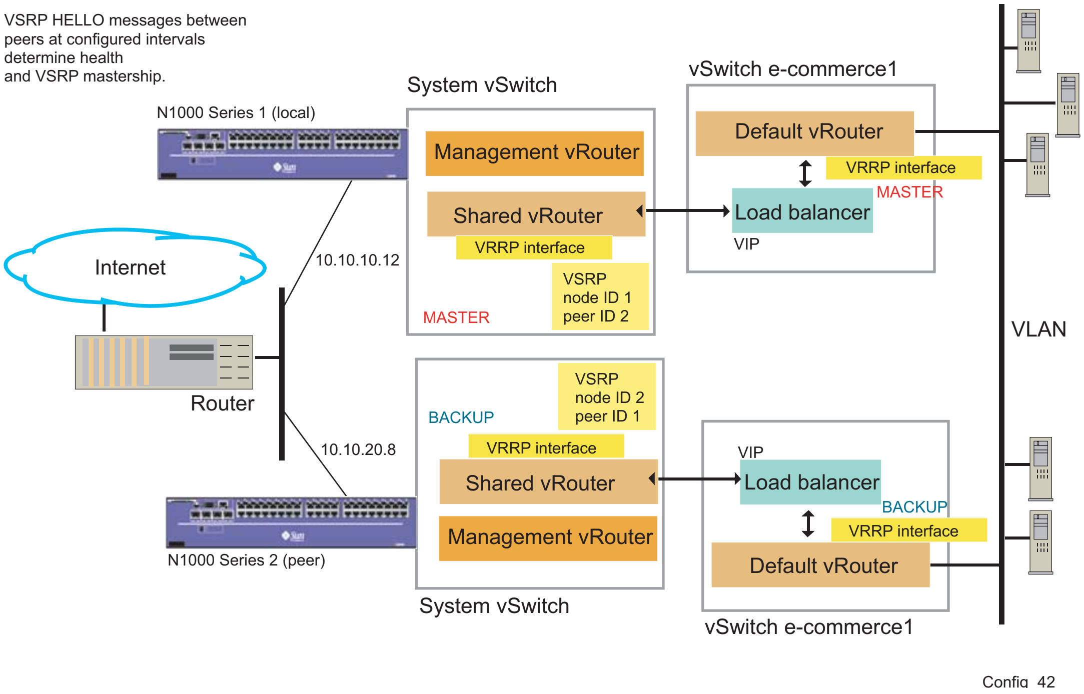
Configuring All or None Mode

When VSRP is configured in the All or None Mode on redundant Sun Secure Application Switches, the switches exchange state and health information (using TCP) to determine whether a switch should run in a master (active) or backup (standby) state. If the master experiences a failure, virtual service traffic switches over to the VSRP backup, or peer node.

With VRRP and VSRP running on redundant Sun Secure Application Switches, you can configure VRRP to accept “hints” from VSRP in electing a VRRP master. You do this with the `vsrpPreference` option in the VRRP configuration. This preference setting helps guarantee the most efficient traffic routing. See “Configuring VRRP” on page 26. You can also configure VSRP to accept hints from VRRP in electing a VSRP master. You do this by setting the `vrrpPreference` option in the VSRP configuration. See Step 3 on page 34.

When first configured, VSRP remains in a backup, or hold-down state until the first VSRP election takes place. During this period, the configured virtual services are deactivated. Once the election identifies the VSRP master, the virtual services are reactivated on the master.

In the sample network illustrated in the following figure, two physical N2000 Series switches are configured as VSRP peers where N1000 Series 1 is the VSRP master and N1000 Series 2 is the VSRP backup.



Perform the following configuration steps to configure VSRP in the sample network.

1. **Configure the local Sun Secure Application Switch by entering the node identifier.**

```
sun(config)#
sun(config)# redundancy
sun(config-redundancy vsrp)# node nodeID 1
```

Note – The default setting for `electionMode` is `allorNone`. You will not have to specify `allorNone` mode for this node unless you have reset the default to `perVswitch`.

---

2. **On the same Sun Secure Application Switch, configure the peer node by specifying the peer numeric identifier.**

```
sun(config-redundancy vsrp)# node nodeId 1 peer peerId 2 port 4535
```

If you do not configure an `electionPreference`, the local and peer nodes use their respective Node IDs to determine which system should be the elected master. The Sun Secure Application Switch with the higher `nodeID` value becomes the master node.

3. **Configure the session over which the local and peer nodes exchange TCP messages.**

```
sun(config-redundancy vsrp)# node nodeId 1 peer peerId 2 session
system:shared 10.10.20.8
```

You can configure multiple sessions for added resiliency. The elected switch is only considered “down” by the backup switch if all sessions to the elected switch fail.

4. **Specify the VRRP Preference option if you want VRRP and VSRP to be active on the same physical switch.**

The VRRP Preference is used to calculate the Current Election Preference. The master (active) switch is the switch with the highest Current Election Preference.

The following CLI session sets the VRRP Preference at 200, which provides this switch with an Elected State of master (active).

```
sun(config-redundancy vsrp)# node 1 vrrppreference 200
sun(config-redundancy vsrp)# show node


Node ID:                     1
Port:                        4121
Election Mode:               allOrNone
Election Preference:         100
Elected Increase:            100
Hello Time:                  1
Missing Hello Count:         3
Admin State:                 enabled
VRRP Preference:             200
Oper Status:                 down
Elected State:               master
Current Election Preference: 400
Election Changes:            2
Number Master Vrrp Routers:  1
```

Configuring Per vSwitch Mode

When VSRP is configured in the Per vSwitch Mode on redundant Sun Secure Application Switches, both switches can simultaneously host active vSwitches and standby vSwitches. Allowing active and standby vSwitches to exist in one switch better utilizes the standby switch so that it is not completely idle while providing redundancy to the active switch. Both switches provide virtual services to the data center and both switches participate in Layer 2 and Layer 3 redundancy. If either physical switch is unable to maintain the active virtual services, the active vSwitches on the affected physical switch will transition to the standby state and the standby vSwitches will transition to the active state.

Perform the following configuration steps to configure VSRP in the Per vSwitch Mode.

1. **Configure the local node by entering the VSRP electionMode and the node identifier.**

```
sun(config)# redundancy
sun(config-redundancy vsrp)# node nodeID 1 electionMode pervSwitch
sun(config-redundancy vsrp)# show node
Node ID:                      1
Port:                         4121
Election Mode:                perVSwitch
Election Preference:         100
Elected Increase:            100
Hello Time:                  1
Missing Hello Count:         3
Admin State:                 enabled
VRRP Preference:             200
Oper Status:                 N/A
Elected State:               perVSwitch
Current Election Preference: N/A
Election Changes:            N/A
Number Master Vrrp Routers:  N/A
sun(config-redundancy vsrp)#
```

2. **Specify the VSRP Election Preference for each vSwitch.**

   The vSwitch with the highest Election Preference becomes the master (active) switch. If all vSwitches have the same Election Preference value, the vSwitch with the higher node ID becomes master (active).

   The following CLI session sets the VSRP Election Preference at 400, which makes it the master vSwitch.

```
sun(config)# vSwitch erp
sun(config-vSwitch-erp)# redundancy
sun(config-vSwitch-erp redundancy)# vsrp electionPreference 400
sun(config-vSwitch-erp redundancy)# show vsrp

Election Preference:         400
Elected State:               master
Current Election Preference: 400
Election Changes:            0
```

You can also view the VSRP settings of the backup switch.

```
sun(config-vSwitch-erp redundancy)# exit
sun(config-vSwitch-erp)# exit
sun(config)# vSwitch-erp-1
sun(config-vSwitch-erp-1)# redundancy
sun(config-vSwitch-erp-1 redundancy)# show vsrp

Election Preference:         200
Elected State:               backup
Current Election Preference: 200
Election Changes:            0
```

CHAPTER **6**

How to Configure Server Health Checks

Server health checks (SHC) allow you to verify the content accessibility of large web sites. As content expands and various information is distributed across a variety of servers, health checks are crucial to ensure end-to-end availability. Health checks monitor the state of servers in a real service group to ensure their availability for load balancing. This chapter covers the configuration tasks that you will perform to implement server health checks.

For additional information about configuration tasks, and the specific commands cited in this chapter, see the *Sun Secure Application Switch – Command Reference for V4.0*.

Note – Examples throughout this chapter show how to perform configuration tasks using the command-line interface (CLI). For information about using the other methods available, see the *Sun Secure Application Switch – Getting Started Guide*.

About Server Health Checks

The Sun Secure Application Switch supports out-of-band, in-line, and passive server health checks. Based on the health check parameters you set, the switch determines which servers in a service group are available for load balancing and creates an availability table. If a server is unresponsive, the switch removes the server from the table. If the server responds to subsequent polls, the switch returns the server to the table.

Note – When distributing the traffic load, the switch favors servers with faster response times to health check probes.

For more information about the types of server health checks, see the *Sun Secure Application Switch – Command Reference for V4.0* or the *Sun Secure Application Switch – Online Help for V4.0*.

The following are key factors to consider before you configure server health checks:

- **`realService weight` parameter** – The switch uses the configuration setting of this parameter to distribute the traffic load.
  - When the `realService weight` is set to `dynamic`, the switch calculates latency values to determine the load-balance weights for the real services.
  - When the `realService weight` is set to a specific weight, the switch uses the health check results only to determine whether the server is available.
- **Server memory and resources** – Consider the available resources before running health checks that could impact performance.
  - Certain types of health check probes consume more switch and server resources (such as memory). For example, HTTP and LIST probe types consume more switch and server memory than TCP and ICMP probe types.
  - The mode of certain health checks can consume more server resources than other modes. For example, the `userlogin` mode of an FTP probe requires more transactions than other modes. See the *Sun Secure Application Switch – Command Reference for V4.0* for more information about the modes of the various probe types.

You can configure one server health check for each service group defined, but the same health check can be used for more than one service group. The following sections show examples of how to configure each type of health check.

Configuring Out-of-Band Server Health Checks

Out-of-band server health checks require the following settings:

- Health check profile name
- Application-specific probe type
- Associated service group

You may need to configure additional settings depending on the probe type defined. For more information about health check profile settings and the available probe types, see the *Sun Secure Application Switch – Command Reference for V4.0*.

Perform the following steps to configure an out-of-band server health check:

1. **Create a named health check profile.**
2. **Define the probe type and configure the settings for the defined probe type.**
3. **Apply the health check profile to a service group.**

The following CLI session creates the health check profile hc1, which uses probe type TCP with a maximum of five retries before determining that a unresponsive server is unavailable.

```
sun(config-vSwitch-e-commerce loadBalance)# healthCheckProfile hc1
type TCP retries 5
```

To display the properties of the health check profile, use the show healthCheckProfile command.

```
sun(config-vSwitch-e-commerce loadBalance)# show healthCheckProfile
 Name:               hc1
 Type:               TCP
 Interval:           5
 Retries:            5
 Success Rate:       0
 Timeout:            2
 Count:              3
```

The following CLI session applies health check hc1 to service group cgiServers.

```
sun(config-vSwitch-e-commerce2 loadBalance)# serviceGroup cgiServers
healthName hc1
```

To display the settings of the service group, use the show serviceGroup command.

```
sun(config-vSwitch-e-commerce2 loadBalance)# show serviceGroup

Name:                                 cgiServers
Load Balance Type:                    roundRobin
Configured Real Services:             rs4; rs5; rs6
Active Real Services:                 rs4; rs5; rs6
Admin State:                          enabled
Virtual Service:                      N/A
Health Check Name:                    hc1
Source Address Mask:                  255.255.255.255
In-Line Health Check:                 enabled
Oper Status:                          inactive
```

In-line Server Health Checks

In-line server health checks determine server health without waiting for the next out-of-band-polling interval. If an in-line health check determines that a server is unresponsive because of a failed TCP connection or HTTP request, the switch removes the server from the service group. The switch places the removed server back into the load-balancing rotation when out-of-band health checks probe the server to determine that the server is again available.

Passive Server Health Checks

Passive server health checks monitor the in-line health checks for successful server activity. If, according to the in-line checks, the server is responsive and healthy, the passive health check determines that the normal polling of the out-of-band probes can be slowed down. This ensures more efficient use of switch and server resources.

By default, passive server health checks are off. To enable this feature, set the server health check profile `SuccessiveCount` parameter to a number greater than 0. All servers in the service group must be responsive and reporting good health for the passive health check feature to operate.

Note – Passive server health check settings cannot be configured during the initial creation of the health check profile. You must configure these parameters once a health check profile exists.

CHAPTER 7

Firewall Load Balance Configuration

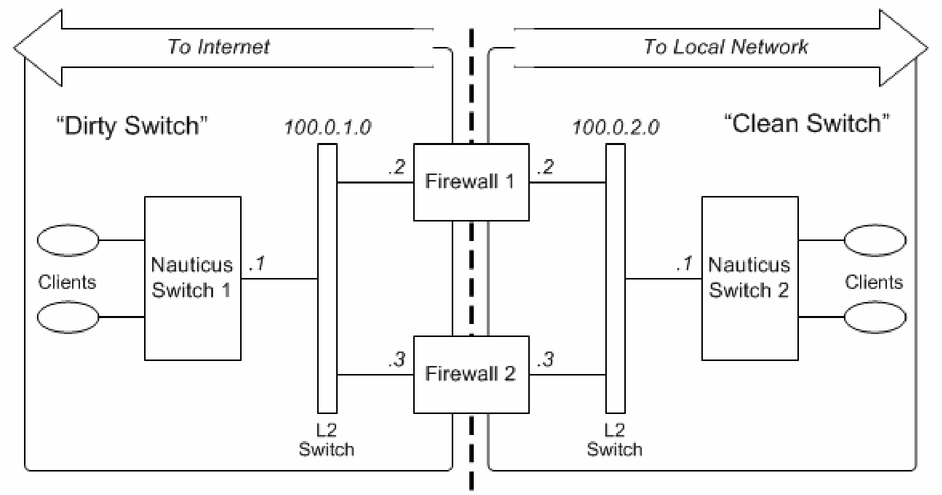
The firewall devices supported for use with the San Secure Application Switch are: Encasement, Checkpoint NSG and NGX, and Cisco PIX.

The instructions below describe how to configure the Sun Secure Application Switch for Firewall Load Balancing applications. If serves need to communicate to clients, it is assumed that NAT is already configured and running properly.

Configuring firewall load balancing is comprised of the following tasks:

- creating and configuring the clean switch
- creating and configuring the dirty switch
- creating Health Check Profiles
- creating the Firewall Load Balancing Virtual Service Group
- verifying the configuration

Firewall load balance configuration may be activated using either the GUI or the CLI. The configuration example that follows assumes you are configuring the firewall according to the illustration below. Note that in this configuration the dirty switch interfaces with the Internet.



Creating the Dirty Switch Using the GUI

1. **Click vSwitch**
2. **Click Add**
3. **Type the name of the virtual switch into the Name field.**

   In this example, the name will be dirty_ switch. You can provide additional information about the virtual switch in the Description field.

4. **Click the Submit button.**

Configuring the Dirty Switch Using the GUI

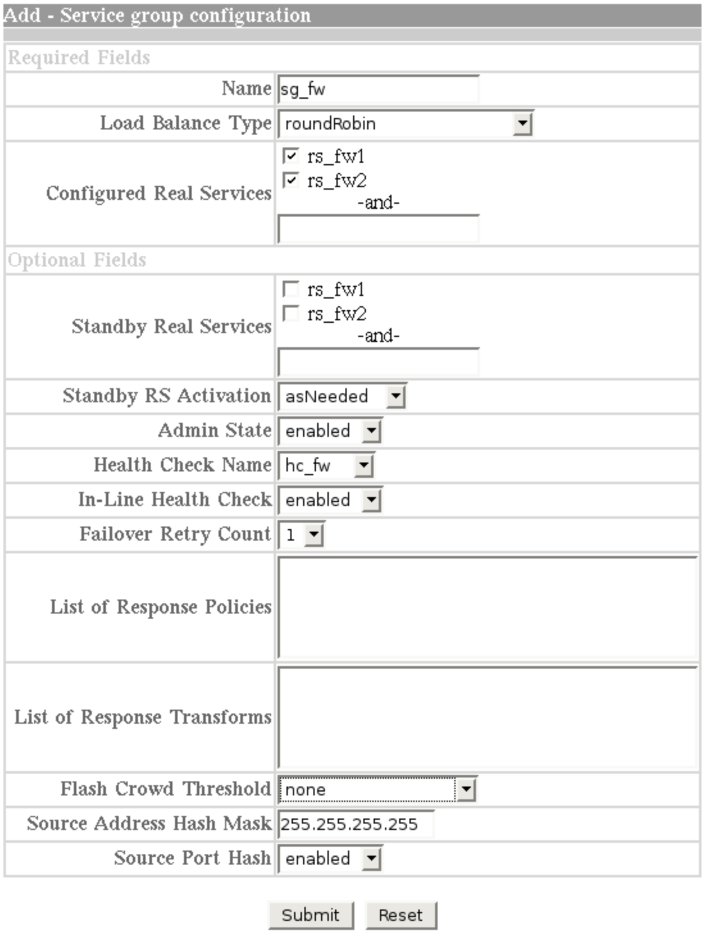
1. **Click dirty_switch.**
2. **Click Load Balance.**
3. **Click Host.**
4. **Click Add.**
5. **Type the firewall name into the Name field.**

   For example, firewall_dirty.
6. **Enter the IP Address for switch.**

   In this example, you would use the IP Address for dirty_switch.
7. **Click the Submit button**
8. **Click Real Service.**
9. **Click Add.**
10. **Type the name of the real service into the Name field.**
11. **From the Host Name drop-down menu, select the host associated with the Real Service.**
12. **Select Ignore from the Port drop-down menu.**
13. **Click the Submit button located at the bottom of the screen.**
14. **Click Service Group.**
15. **Click Add.**

16. **Type the firewall name into the Name field.**

    For example, if this particular firewall is to be associated with a corporate service group, you could name it sg_fw.



Add - Service group configuration

Required Fields	
Name	sg_fw
Load Balance Type	roundRobin
Configured Real Services	☑ rs_fw1 ☑ rs_fw2 -and-
Optional Fields	
Standby Real Services	□ rs_fw1 □ rs_fw2 -and-
Standby RS Activation	asNeeded
Admin State	enabled
Health Check Name	hc_fw
In-Line Health Check	enabled
Failover Retry Count	1
List of Response Policies	
List of Response Transforms	
Flash Crowd Threshold	none
Source Address Hash Mask	255.255.255.255
Source Port Hash	enabled

17. **Select a load balance type from the Load Balance drop-down menu.**

    Round Robin and Weighted Hash are both supported load balance types. Select one of the two supported load balancing methods for this field.

18. **Select the Configured Real Services by selecting the appropriate check boxes.**

    A real service can be either Configured or Stanby, but not both. You must have a minimum of two configured real services to use load balancing configuration.

19. **Use all other default settings and click the Submit button.**

Creating the Health Check Profile Using the GUI

Note – It is assumed that Echo Responder is already configured on the switch(es) you'll be using in conjunction with firewall load balancing.

---

1. **Click Health Check Profile.**
2. **Click Add.**
3. **Type a name for the health check in the Name field.**
4. **Select either Echo TCP, Echo UDP, or ICMP from the Type drop-down menu.**

   Select Echo TCP for TCP handshake, select Echo UDP for stateless, and select ICMP for ping.

5. **Click the Next button to continue.**
6. **On the screen presented, select the destination Address.**

   Since you are configuring the dirty switch, you should enter the IP for the clean switch.

7. **Click the Submit button.**
8. **Click System.**
9. **Click vSwitch dirty-switch.**
10. **Click Load Balance.**
11. **Click Service Group.**
12. **Select the Service Group associated with the firewall, then click Modify.**
13. **Select a health check from the Health Check Name drop-down menu.**

    Typically, health checks verify connectivity through the firewall.

14. **Use all other default settings and click the Submit button to continue.**

Creating the Firewall Load Balance Virtual Service Group Using the GUI

1. **Click Virtual Service.**
2. **Click Add.**
3. **Type the name of the virtual server.**
4. **Select FWLB from the Service Type drop-down menu.**
5. **Click the Next button to continue.**
6. **Select the Service Group from the drop-down menu.**

   Fill in the optional description field to provide additional identifying information about the service group.
7. **Use all other default settings and Click the Submit button to continue.**

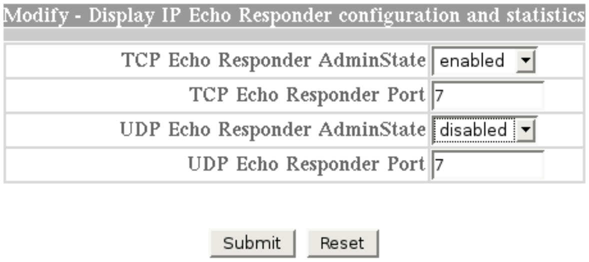
   At this point, the OperStatus should display as Active.

Configuring the Echo Responder on the Clean Switch Using the GUI

1. **Click System vRouter.**
2. **Click Shared.**
3. **Click IP.**
4. **Click Echo Responder.**
5. **Click Modify.**

6. **From the drop-down menu, enable either TCP Echo Responder Admin State or UDP Echo Responder Admin State.**

   The default setting for the for the responder port is seven; you can change it to another value.



Modify - Display IP Echo Responder configuration and statistics

TCP Echo Responder AdminState	enabled
TCP Echo Responder Port	7
UDP Echo Responder AdminState	disabled
UDP Echo Responder Port	7

7. **Click the Submit button.**

   The Op State field should display Up. If Initialization is displayed, wait 30 seconds, click Refresh, and check the display.

Creating the Clean Switch Using the GUI

1. **Click System.**
2. **Click vSwitch.**
3. **Click the Add icon.**
4. **Type the name of the virtual switch into the Name field.**

   In this example, the name will be clean_ switch.
5. **Click the Submit button to continue.**

Configuring the Clean Switch Using the GUI

1. **Click clean_switch.**
2. **Click Load Balance.**
3. **Click Add.**
4. **Type the name of the real service into the Name field.**
5. **Select Ignore from the Port drop-down menu.**
6. **Click the Submit button located at the bottom of the screen.**
7. **Click Service Group.**
8. **Click Add.**
9. **Type the firewall name into the Name field.**
10. **Select a load balance type from the Load Balance drop-down menu.**

    Round Robin and Weighted Hash are both supported load balance types. Select one of the two.
11. **Select the Configured Real Services by selecting the appropriate check boxes.**

    A real service can be either Configured or Stanby, but not both.
12. **Click the Submit button to continue.**

Creating the Health Check Profile Using the GUI

Note – It is assumed that Echo Responder is already configured on the switch(es) you'll be using in conjunction with firewall load balancing.

---

1. **Click Load Balance.**
2. **Click Health Check Profile.**
3. **Click Add.**
4. **Type a name for the health check in the Name field.**
5. **Select either Echo TCP, Echo UDP, or ICMP from the Type drop-down menu.**

    Select Echo TCP for TCP handshake, select Echo UDP for stateless, and select ICMP for ping.
6. **Click the Next button to continue.**

7. **On the screen presented, select the destination Address.**

   Since you are configuring the clean switch, you should enter the IP for the dirty switch.

8. **Click the Submit button.**
9. **Click System.**
10. **Click vSwitch dirty-switch.**
11. **Click Load Balance.**
12. **Click Service Group.**
13. **Select the Service Group associated with the firewall, then click Modify.**
14. **Select a health check from the Health Check Name drop-down menu.**

    Typically, health checks verify connectivity through the firewall.

15. **Use all other default settings and click the Submit button to continue.**

Creating the Firewall Load Balance Virtual Service Using the GUI

1. **Click Virtual Service.**
2. **Click Add.**
3. **Type the name of the virtual server.**
4. **Select FWLB from the Service Type drop-down menu.**
5. **Click the Next button to continue.**
6. **Select the Service Group from the drop-down menu**

   Fill in the optional description field to provide additional identifying information about the service group.

7. **Use all other default settings and Click the Submit button to continue.**

Configuring the Echo Responder on the Dirty Switch Using the GUI

1. **Click System vRouter.**
2. **Click Shared.**
3. **Click IP.**
4. **Click Echo Responder.**
5. **Click Modify.**
6. **From the drop-down menu, enable either TCP Echo Responder Admin State or UDP Echo Responder Admin State.**

   The default setting for the for the responder port is seven; you can change it to another value.

7. **Click the Submit button.**

   The Op State field should display Up. If Initialization is displayed, wait 30 seconds, click Refresh, and check the display.

Verifying Firewall Load Balance Configuration Using the GUI

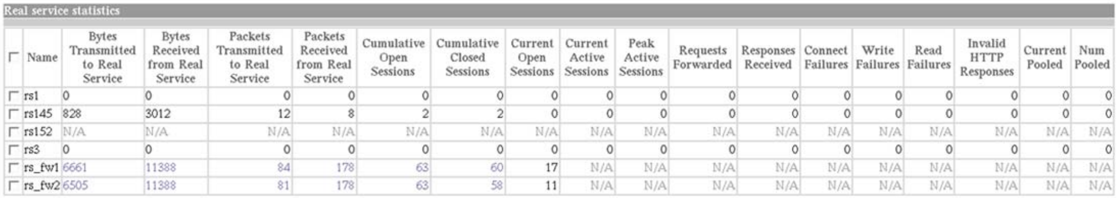
Note – For round robin you have to use a second source address, or client, in order to load balance among the firewalls.

---

1. **Send a request to your VIP from a UNIX, Linux, or PC station.**
2. **Go to vSwitch dirty_switch.**
3. **Click Load Balance.**
4. **Click Real Service.**
5. **Click the plus sign adjacent to Real Service to display the submenu.**

6. **Click Statistics**

   The data in Cumulative Open Session and Cumulative Closed Session should be almost equal.



Real service statistics

☐	Name	Bytes Transmitted to Real Service	Bytes Received from Real Service	Packets Transmitted to Real Service	Packets Received from Real Service	Cumulative Open Sessions	Cumulative Closed Sessions	Current Open Sessions	Current Active Sessions	Peak Active Sessions	Requests Forwarded	Responses Received	Connect Failures	Write Failures	Read Failures	Invalid HTTP Responses	Current Pooled	Num Pooled
☐	rs1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
☐	rs145	828	3012	12	8	2	2	0	0	0	0	0	0	0	0	0	0	0
☐	rs152	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
☐	rs3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
☐	rs_fw1	6661	11388	84	178	63	60	17	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
☐	rs_fw2	6505	11388	81	178	63	58	11	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Creating the Dirty Switch Using the CLI

1. **Create the dirty switch.**

   ```
   sun(config)# vswitch dirty
   ```

2. **At the prompt press the Y key to continue.**

   Create new vSwitch "dirty"? (y or n): y

3. **Create the host.**

   ```
   sun(config-vSwitch-dirty)# loadbalance host <hostname>
   <IPAddress>
   ```

   For example, if the hostname is fw_dirty and the IPAddress is 100.0.1.4, you would type the following:

   ```
   sun(config-vSwitch-dirty)# loadbalance host fw_dirty 100.0.1.4
   ```

4. **Create the real service for the dirty switch.**

   ```
   sun(config-vSwitch-dirty loadBalance)# realService <realService name>
   <host name> rs_fw_dirty fw_dirty port ignore.
   ```

   For example, if the real service name is rs_fw_dirty and the host name is fw_dirty you would type

   ```
   sun(config-vSwitch-dirty loadBalance)# realService rs_fw_dirty fw_dirty
   port ignore.
   ```

5. **Create the service group.**

   ```
   sun(config-vSwitch-dirty loadBalance)# serviceGroup <service group
   name> roundRobin <real service name>
   ```

   For example, if the service group name is fw_sg1 and the real service name is rs_fw_dirty, you would type:

   ```
   sun(config-vSwitch-dirty loadBalance)# serviceGroup fw_sg1
   roundRobin rs_fw_dirty
   ```

6. **Create the health check.**

   Note – In this example, you will be using an ECHO_TCP health check.

   ```
   sun(config-vSwitch-dirty loadBalance)# healthCheckprofile <health
   check profile name> ECHO_TCP destinationaddress <IP Address>
   ```

   For example, if the health check name is fw_tcp and the IP address is 100.64.27.9, you would type:

   ```
   sun(config-vSwitch-dirty loadBalance)# healthCheckprofile fw_tcp
   ECHO_TCP destinationaddress 100.64.27.9
   ```

7. **Modify the Service Group to include the health check you just created.**

   ```
   sun(config-vSwitch-dirty loadBalance)# serviceGroup <service group
   name> healthName <health check name>
   ```

   For example, if the service group is fw_sg1 and the health check name is fw_tcp, you would type:

   ```
   sun(config-vSwitch-dirty loadBalance)# serviceGroup fw_sg1
   healthName fw_tcp
   ```

8. **Create the Virtual Service.**

   ```
   sun(config-vSwitch-dirty loadBalance)# virtualService <virtual service
   name> <service type> <service group name>
   ```

   For example, if the virtual service name is vs_fw, the service type is firewall load balancing, and the service group name is fw_sg1, you would type:

   ```
   sun(config-vSwitch-dirty loadBalance)# virtualService vs_fw FWLB
   fw_sg1
   ```

9. **Configure the Echo responder on the dirty switch.**

Note – The echo responder created on the dirty switch is used by health checks issued on the clean switch.

`sun(config-vSwitch-system vRouter-shared ip)#` **echoResponder TCpEchoResponderAdminState enabled**

Creating the Clean Switch Using the CLI

1. **Create the clean switch.**

   `sun(config)#` **vswitch clean**

2. **At the prompt press the Y key to continue.**

   Create new vSwitch "clean"? (y or n): y

3. **Create the host.**

   `sun(config-vSwitch-clean)#` **`loadbalance host`** `<hostname> <IPAddress>`

4. **Create the real service for the dirty switch.**

   `sun(config-vSwitch-clean loadBalance)#` **realService** <realService name> <host name> rs_fw_dirty fw_clean **port ignore**.

5. **Create the service group.**

   `sun(config-vSwitch-dirty loadBalance)#` **serviceGroup** <service group name> **roundRobin** <real service name>

6. **Create the health check.**

Note – `In this example, you will be using an ECHO_TCP health check.`

`sun(config-vSwitch-dirty loadBalance)#` **healthCheckprofile** <health check profile name> **ECHO_TCP destinationaddress** <IP Address>

7. **Modify the Service Group to include the health check you just created.**

   `sun(config-vSwitch-dirty loadBalance)#` **serviceGroup** <service group name> **healthName** <health check name>

8. **Modify the Service Group to include the health check you just created.**

   `sun(config-vSwitch-dirty loadBalance)#` **serviceGroup** <service group name> **healthName** <health check name>

9. **Create the Virtual Service.**

   `sun(config-vSwitch-dirty loadBalance)#` **virtualService** <virtual service name> <service type> <service group name>

10. **Configure the Echo responder on the dirty switch.**

Note – The echo responder created on the dirty switch is used by health checks issued on the clean switch.

`sun(config-vSwitch-system vRouter-shared ip)#` **echoResponder TCpEchoResponderAdminState enabled**

Verifying the Firewall Load Balance Configuration Using CLI

1. **To very the firewall load balance configuration, issue the following command:**

   ```
   sun(config#) vSwitch dirty load balance real service statistics
   ```

2. **Issue the command a second time and compared the captured data.**

Creating a Passthrough Health Check Using the CLI

1. **Create a new health check profile.**

   `sun(config)#` **healthcheckprofile fw_icmp**

2. **Associate the health check with the destination address.**

   `sun(config)#` **healthcheckprofile fw_icmp advanced passthroughMode enabled destinationaddress** <IP Address>

CHAPTER 8

Secure Sockets Layer (SSL) Certificates for Configuration Synchronization

The Secure Sockets Layer (SSL) makes use of TCP as a communication protocol, providing a secure and reliable end-to-end connection between two points over a network. The SSL protocol runs above the TCP transport layer and below higher-level protocols like HTTP. The main objectives of SSL include: authenticating the client and server to each other, ensuring data integrity, and securing data.

- SSL server authentication allows a user to confirm a server's identity.
- SSL client authentication allows a server to confirm a user's identity.
- An encrypted SSL connection requires all information that is sent between a client and a server to be encrypted by the sending software and decrypted by the receiving software, thus providing a high degree of confidentiality.

SSL is used to establish a secure link between peer switches in Configuration Synchronization applications. Before such applications can be established, you must fist configure RSA certificate keypairs.

Configure SSL Parameters

The procedure to configure SSL parameters includes five separate tasks. You must perform the five tasks outlined in the table below.

TABLE 8-1 Task Map for Establishing SSL

Task	Description
1	Create the SSL certificates
2	Export the SSL certificates
3	Import the SSL certificates
4	Configure the local SSL certificates
5	Configure the remote SSL certificates

While this list provides an overview of the tasks in this procedure, these tasks are broken down in more detail within this chapter. For additional information about specific commands cited in this chapter, refer to the *Sun Secure Application Switch – Command Reference*.

Create the SSL Certificates

You must create certificates on the system vSwitch. In addition, you must use the RSA algorithm, as DSA is not supported. When creating a certificate, you must define the bit length, the number of days the certificate is valid, and provide the fully qualified domain name (FQDN) that will be associated with your SSL application. A certificate that is created on a switch is referred to as a local certificate.

Create the SSL Certificate on the Source Switch

1. **Create the certificate on the Source switch's system vSwitch and define the bit length.**

   The sourcecertname is the variable representing the name of the certificate on the Source switch.

   ```
   sun(config)# vswitch system ckm generate <sourcecertname> bitLength
   <512|1024|2048> algorithm rsa
   ```

   For example, if the name of the certificate for the Source switch is `cert1` and the bit length is `1024`, you would type the syntax below.

   ```
   sun(config)# vswitch system ckm generate cert1 bitLength 1024
   algorithm rsa
   ```

2. **Provide the fully qualified domain name (FQDN) and define the number of days the certificate will be valid before it expires.**

   The certificate can be valid from 1 to 730 days.

   ```
   sun(config)# vswitch system ckm csr <sourcecertname> <www.yourfqdn.com>
   ca true makeTestCert true testCertDays <1...730>
   ```

   For example, if the fully qualified domain name is `switch1.com` and you want the certificate to be valid for `365` days, type the syntax below.

   ```
   sun(config)# vswitch system ckm csr cert1 www.switch1.com ca true
   makeTestCert true testCertDays 365
   ```

   As the certificate is generated, it appears on the monitor.

Create the SSL Certificate on the Target Switch

1. **Create the certificate on the Target switch's system vSwitch and define the bit length.**

   The targetcertname is the variable representing the name of the certificate on the Target switch.

   ```
   sun(config)# vswitch system ckm generate <targetcertname> bitLength
   <512|1024|2048> algorithm rsa
   ```

2. **Provide the fully qualified domain name (FQDN) and define the number of days the certificate will be valid before it expires.**

   The certificate can be valid from one to 730 days.

   ```
   sun(config)# vswitch system ckm csr <targetcertname>
   <www.yourfqdn.com> ca true makeTestCert true testCertDays <1...730>
   ```

   As the certificate is generated, it appears on the monitor.

Export the Local SSL Certificate

A certificate that is created on a switch is referred to as a local certificate. You must export the local certificate created on the Source switch. Later in this procedure you will import the local certificate created on the Source switch to the Target switch.

Export the Local SSL Certificate from the Source Switch

1. **When exporting the local certificate, you must use the certificate name specified earlier in this procedure. Using the CLI for the Source switch, type the syntax below.**

   ```
   sun(config)# vswitch system ckm export <sourcecertname> certificate
   ```

   In the UNIX terminal window, highlight and copy the certificate.

Note – When you highlight the certificate you must include everything, including the Begin Certificate text and End Certificate text.

```
-----BEGIN CERTIFICATE-----
MIICBTCCAW6gAwIBAgIBATANBgkqhkiG9w0BAQUFADAaMRgwFg
YDVQQDEw93d3cuc3dpdGNoMS5jb20wHhcNMDYxMDIwMjA0NjQx
WhcNMDcxMDIwMjA0NjQxWjAaMRgwFgYDVQQDEw93d3cuc3dpdG
NoMS5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJ
AoGBALjQR4f5j9VD5JMLr9hQ1xMQbAdO0uI+7NxRoPkccPvNh+
pGEepZ5MRWaWd7WzYjnmD1C2fsIMNnQdOBuwb90I4p3j1L3C+E
rHFXfaEDau5BjO0HDzm/h4pzyodEeTkkbIwXAk638MmwcJGORZ
C6KxP/hLNNG0Zpzubk1EMFaKAHAgMBAAGjWzBZMBoGA1UdEQQT
MBGCD3d3dy5zd2l0Y2gxLmNvbTAPBgNVHRMBAf8EBTADAQH/MA
sGA1UdDwQEAwIBtjAdBgNVHQ4EFgQUCh4EIiW0TSGk91cRF+Cq
5zyXm3QwDQYJKoZIhvcNAQEFBQADgYEAKO2sdq4XHwIebJc67U
VtXD5xOEzkpWVxuzmzDS2i/IkaEOSvoMx4uBE02xIfvuodWJWm
sV8cEVpTIxJHGUSO4FQUUVkv6dm6q5Ka8JG8kjlAbFHyN70AzY
6jigl5wT4Vum7FgnFCHYV2Sk7HHlIZSljBgDWBWLifjrUqVpEX
hcI=
-----END CERTIFICATE-----
```

Import the Remote SSL Certificate

A certificate that is imported to a switch is referred to as a remote certificate. Importing the remote certificate is a crucial step in establishing SSL communication. To establish SSL communication, the local certificate exported from the Source switch must be imported to the Target switch. Likewise, the local certificate exported from the Target switch must be imported to the Source switch. This enables one switch to encrypt information going to the other switch, establishing a secure point-to-point connection. More importantly, each switch is able to verify the identify of the other one.

Import the SSL Certificate From the Source Switch To the Target Switch

1. **Using the Target switch's CLI, type the following syntax.**

   ```
   sun(config)# vswitch system ckm import paste <sourcecertname>
   certificate internalCkm
   ```

   In the UNIX terminal window, paste the certificate in place, then press ctrl-z.

Export the Private Key From the Source Switch

After you import the Source switch's local certificate to the Target switch, you must export the Source switch's private key.

1. **When exporting the private key, you must specify a password, between 4 and 255 characters long, that will be used to encrypt information. Using the CLI for the Source switch, type the syntax below.**

   ```
   sun(config)# vswitch system ckm export <sourcecertname> privateKey
   password <yourpassword>
   ```

For example, if the name of the source certificate is *cert1* and your password is *changeme*, type the syntax below.

```
sun(config)# vswitch system ckm export cert1 privateKey password changeme
```

In the UNIX terminal window, highlight and copy the Private Key.

Note – When you highlight the Private Key you must select everything, including the Begin Internal CKM Private Key text and End Internal CKM Private Key text.

```
-----BEGIN INTERNAL CKM PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,8EA17E7DEAC775AD

EnCs56OmlleuV0eoEUMVW9FnJBfS1dr65shFVmPB20H7b/Gv9y
PzUroZm0rgJSdIS6A+HqIrBsjHwAHfpOMr+4P7W11rdH83WMSV
WE7FAII7AncAwyuuciRhsdTBfEriVkzHNahDkpL4y/8Qlsy0QF
S8IaGOQkTNW1IbEmOQ8i3lmZFS/T7MifTNoO/ujg4c0HKcMF1J
wagu9YLUm5PW/8oW1ndkUoYqwFi+o84RMpHUkPmINlAhLwo3bI
WQjDI05HlqoiKPhP4llMMdTJMLXLw+OsSXacijRAujNYHj2gEP
OVf2c8WALLPEsFBed3kd6PAv6ljPVvH1Wkeu4r7O7fUFZFjwho
zBvACZ7LAZwgsfr2y2p1KqgK7vaR0QBnygcDBK+mTPHgfor05M
oGRTmg2Xuqxx/QlBpOn/7X4+az6WCU3CPSjcfsQmxvFkAYPDO8
eYm4F/aI42sC/Oq1dG7gkt1EXPTv0tfm1cel6MbszTmWqeizMU
EcCG/kCV+aBUjvnhakJBguqQD7IUa2PaONECMJbsmte1vw8V0k
6uaRaOyvv8sLoDx00Y/lq6Fru7oMHEMMqGzxevVVzGPmgndyXZ
KhNgVRc7cuNoU0g/HBxJNiAd3StMlXMBXtTBsepF9U3HzMjt72
1Fq8j7Rmai1fN2UHp3Oq4TcKg3bzoWfkyp7rZPCgwvOEdwlgSj
iBnJOwIxNZzeSMv/VDk5uGE3U5R8KE+AeXGlG/zlZL9Ci7rJRn
m4fSceQCiBhIo+6YuzCWliF5sgQorQJxMu5PLbV/lalyI9t8od
MT7C0jAIismR5X9H1yuTqnevRwD878AG Mum3S+KzBJ4=
-----END INTERNAL CKM PRIVATE KEY-----
```

Import the Private Key From the Source Switch To the Target Switch

1. **Using the CLI for the Target switch, type the following syntax to import the private key.**

   ```
   sun(config)# vswitch system ckm import paste <sourcecertname>
   privateKey internalCkm
   ```

   At the prompt to enter data, paste the certificate in the UNIX terminal window, then press Ctrl-z to continue.

Export the Local SSL Certificate from the Target Switch

You must export the local certificate that was created on the Target switch. After the certificate is imported, the next task in this procedure will be to import the certificate to the Source switch.

1. **When exporting the local certificate, you must use the certificate name specified previously in this procedure.**

   Using the CLI for the Target switch, type the following syntax to export the certificate.

   ```
   sun(config)# vswitch system ckm export <targetcertname> certificate
   ```

   In the UNIX terminal window, highlight and copy the certificate.

Import the SSL Certificate From the Target Switch To the Source Switch

The certificate exported from the Target switch and imported to the Source switch is called a remote certificate.

1. **Using the CLI for the Source switch, type the following syntax to import the remote certificate.**

   ```
   sun(config)# vswitch system ckm import paste <sourcecertname>
   certificate internalCkm
   ```

   In the UNIX terminal window, paste the certificate in place, then press Ctrl-z.

Export the Private Key From the Target switch

After the Target switch's local certificate is imported to the Source switch, you must export the Target switch's private key.

1. **When exporting the private key, you must specify a password, between four and 255 characters long, that will be used to encrypt information. Using the CLI for the Target switch, type the syntax below.**

   ```
   sun(config)# vswitch system ckm export <targetcertname> privateKey
   password <yourpassword>
   ```

   In the UNIX terminal window, highlight and copy the private key.

Import the Private Key From the Target Switch To the Source Switch

1. **Using the CLI for the Source switch, type the following syntax to import the private key.**

   ```
   sun(config)# vswitch system ckm import paste <sourcecertname>
   privateKey internalCkm
   ```

   At the prompt to enter data, paste the certificate in the UNIX terminal window, then press Ctrl-z to continue.

Configure the Local SSL Certificate

A switch may contain more than one local certificate. After you have created the local certificates for use in your SSL communication and you have imported the remote certificates on each switch, you must specify which local certificate to use.

Configure the Local SSL Certificate on the Source switch

```
redundancy configSync localCertName <sourcecertname>
```

Configure the Local SSL Certificate on the Target switch

1. **To configure the local SSL certificate on the Target switch, type the syntax below.**

   ```
   sun(config)# redundancy configSync localCertName <targetcertname>
   ```

Configure the Remote SSL certificate

A switch may contain more than one remote certificate. After the local certificates for use in your SSL communication have been created and the remote certificates have been imported on each switch, you must specify which remote certificate to use.

Configure the Remote SSL Certificate on the Source switch

1. **To configure the remote SSL certificate on the Source switch, type the syntax below.**

```
sun(config)# redundancy configSync remoteCertName <targetcertname>
```

Configure the Remote SSL Certificate on the Target switch

1. **To configure the remote SSL certificate on the Source switch, type the syntax below.**

```
sun(config)# redundancy configSync remoteCertName <sourcecertname>
```

CHAPTER **9**

Configuration Synchronization

Configuration Synchronization is a set of network management functions that you can use to help maintain a pair of redundant switches. In redundant switch applications, load balance configuration parameters are frequently adjusted. Since many load balancing changes on one switch (the Source switch) must also be performed on a second switch (the Target switch), maintaining this configuration can become complicated and time consuming. Configuration Synchronization alleviates these issues by reducing the number of steps that are required to propagate changes from the Source switch to the Target switch. For additional information about specific Configuration Synchronization commands cited in this chapter, see the *Sun Secure Application Switch – Command Reference for V4.0*.

About Configuration Synchronization

You can establish Configuration Synchronization on the Sun Secure Application Switch using one of three modes: Auto, Manual or Local. If you set the mode to Disable, the TCP connection for the physical switch is turned off and Configuration Synchronization will not function.

To establish Configuration Synchronization, you must set up a Source switch and define a Target switch. For Configuration Synchronization to function, you must properly configure the Local and Remote SSL Certificates.

Note – Only the commands listed on the following table are compatible with Configuration Synchronization.

TABLE 9-1 Commands Compatible with Configuration Synchronization

	vSwitch -- context
	vSwitch ckm import
User Extensible Configuration	vSwitch loadBalance advanced http derivedVariable -- context vSwitch loadBalance advanced http header -- context vSwitch loadBalance advanced http parsedList -- context vSwitch loadBalance advanced http regularExpression -- context vSwitch loadBalance advanced http parsedVariable -- context
	vSwitch loadBalance cookiePersistence
Server Health Check	vSwitch loadBalance healthCheckProfile -- context vSwitch loadBalance healthCheckProfile advanced vSwitch loadBalance healthCheckProfile passive
	vSwitch loadBalance host
	vSwitch loadBalance objectRule
NAT	vSwitch loadBalance outboundNat dynamic -- context vSwitch loadBalance outboundNat dynamic hostIpRange vSwitch loadBalance outboundNat static -- context
CAT	vSwitch loadBalance proxyIPPool -- context
	Switch loadBalance realService -- context Switch loadBalance realService advanced
	vSwitch loadBalance requestPolicy -- context
	vSwitch loadBalance requestTransform
	vSwitch loadBalance responsePolicy -- context
	vSwitch loadBalance responseTransform
	vSwitch loadBalance serviceGroup -- context
	vSwitch loadBalance sorryData
	vSwitch loadBalance virtualService -- context
	vSwitch loadBalance virtualService advanced
Service Bandwidth	vSwitch resource serviceBandwidth

Auto Mode

If a configuration command that was successfully executed on the Source switch is a candidate for Configuration Synchronization, the command is automatically transferred to the Target switch defined by the `configSyncIPAddress`.

Manual Mode

If a configuration command that was successfully executed on the Source switch is a candidate for Configuration Synchronization, the command is logged within a journal file that is stored in Source switch's Flash memory. Issuing a `push` command on the Source switch transfers the journal file to the Target switch defined by the `configSyncIPAddress`.

Manual mode is useful when you are testing a new configuration. If you test a new configuration in Manual mode, you can refrain from pushing the journal file to the Target switch until you are certain that the configuration works properly.

Local Mode

All Configuration Synchronization commands are executed on the Source switch only. To accept commands from a Source switch in Configuration Synchronization applications, the Target switch must be in Local mode.

Establishing Configuration Synchronization: Auto Mode

1. **Enable the Sun Secure Application Switch.**

```
sun> enable
```

2. **Set the switch to configuration mode.**

```
sun# config
```

3. **Define the IP address of the Target switch that is used in your Configuration Synchronization application.**

Note – In this example, the IP address of the Target switch is 10.8.169.143.

```
sun(config)# redundancy configSync targetIpAddress 10.8.169.143
```

4. **Establish Auto as the mode for the Configuration Synchronization application.**

```
sun(config)# redundancy configSync mode auto
```

5. **Verify the Mode was properly configured.**

```
sun(config)# show redundancy configSync
   Target IP Address:  10.8.169.143
   ConfigSync Mode:    auto
   Operational Status: connected
```

6. **Display a summary of Configuration Synchronization activity.**

```
sun(config)# show redundancy configSync summary
   vSwitch:                     server40
   State:                       enabled
   Sync Attempts:              0
   Sync Succeeded:             0
   Sync Succeeded Last:       N/A
   Rcv Sync Succeeded:        0
   Rcv Sync Succeeded Last: N/A
   Push Attempts:              1
   Push Succeeded:             1
   Push Succeeded Last:       Fri Jul 28 15:32:16 2006

   vSwitch:                     server50
   State:                       enabled
   Sync Attempts:              0
   Sync Succeeded:             0
   Sync Succeeded Last:       N/A
   Rcv Sync Succeeded:        0
   Rcv Sync Succeeded Last: N/A
   Push Attempts:              1
   Push Succeeded:             1
   Push Succeeded Last:       Fri Jul 28 15:32:16 2006
```

```
vSwitch:                    system
State:                      enabled
Sync Attempts:              0
Sync Succeeded:             0
Sync Succeeded Last:       N/A
Rcv Sync Succeeded:         0
Rcv Sync Succeeded Last:  N/A
Push Attempts:              1
Push Succeeded:             1
Push Succeeded Last:       Fri Jul 28 15:32:16 2006

vSwitch:                    test
State:                      enabled
Sync Attempts:              0
Sync Succeeded:             0
Sync Succeeded Last:       N/A
Rcv Sync Succeeded:         0
Rcv Sync Succeeded Last:  N/A
Push Attempts:              0
Push Succeeded:             0
Push Succeeded Last:       N/A
```

7. **Create a new virtual switch named scale, then wait a few moments for the prompt to reappear.**

```
sun(config)# vswitch scale
Create new vSwitch "scale"? (y or n): y
```

8. **Issue the following command and verify that the Configuration Synchronization State is enabled. At this point, Configuration Synchronization is actually established.**

```
sun(config-vSwitch-scale)# show redundancy configSync

     ConfigSyncState:           enabled

     Sync Attempts:              0

     Sync Succeeded:             0

     Sync Succeeded Last:

     Rcv Sync Succeeded:        0

     Rcv Sync Succeeded Last:

     Push Attempts:              0

     Push Succeeded:             0

     Push Succeeded Last:
```

9. **Using the CLI on the Source switch, issue the following command.**

```
Sun(config-vSwitch-scale)# loadBalance
```

10. **Perform a cookie persistence test on the Source switch, then wait for the prompt to reappear.**

```
sun(config-vSwitch-scale loadBalance)# cookiePersistence test
```

11. **Verify that the cookie persistence test was performed on the Source switch.**

```
sun(config-vSwitch-scale loadBalance)# show cookiePersistence
  Name:            test
  Cookie Name:     nnSessionID
  Cookie Domain:   N/A
  Cookie Path:     /
  Cookie Expires: N/A
  Secure:          false
```

12. **Using the CLI on the Target Switch, access the vSwitch that was created.**

Note – Make certain you are using the CLI for the Target switch.

```
sun(config)# vSwitch scale loadBalance
```

13. **Verify that the cookie persistence test was performed on the Target switch.**

```
sun(config-vSwitch-scale loadBalance)# show cookiePersistence
  Name:            test
  Cookie Name:     nnSessionID
  Cookie Domain:   N/A
  Cookie Path:     /
  Cookie Expires: N/A
  Secure:          false
```

14. **Verify the mode was properly configured for the Target switch.**

sun(config)# **show redundancy configSync**

```
Target IP Address:  10.8.169.127
ConfigSync Mode: local
Operational Status: connected
sun(config)#
```

Establishing Configuration Synchronization: Manual Mode

1. **Enable the Sun Secure Application Switch.**

```
sun> enable
```

2. **Set the switch to configuration mode.**

```
sun# config
```

3. **Define the IP address of the Target switch that is being used in your Configuration Synchronization application.**

Note – In this example, the IP address of the Target switch is 10.8.169.143

```
sun(config)# redundancy configSync targetIpAddress 10.8.169.143
```

4. **Establish Manual as the mode for the Configuration Synchronization application.**

```
sun(config)# redundancy configSync mode manual
```

5. **Verify that Manual mode was properly configured.**

```
sun(config)# show redundancy configSync

   Target IP Address:  10.8.169.143

   ConfigSync Mode:    manual

   Operational Status: connected
```

6. **Display a summary of Configuration Synchronization activity.**

```
sun(config)# show redundancy configSync summary

   vSwitch:                      scale

   State:                        enabled

   Sync Attempts:              0

   Sync Succeeded:             0

   Sync Succeeded Last:       N/A

   Rcv Sync Succeeded:        0

   Rcv Sync Succeeded Last: N/A

   Push Attempts:              0

   Push Succeeded:             0

   Push Succeeded Last:       N/A
```

```
vSwitch:                    server40
State:                      enabled
Sync Attempts:             0
Sync Succeeded:            0
Sync Succeeded Last:      N/A
Rcv Sync Succeeded:       0
Rcv Sync Succeeded Last: N/A
Push Attempts:             1
Push Succeeded:            1
Push Succeeded Last:      Fri Jul 28 15:32:16 2006

vSwitch:                    server50
State:                      enabled
Sync Attempts:             0
Sync Succeeded:            0
Sync Succeeded Last:      N/A
Rcv Sync Succeeded:       0
Rcv Sync Succeeded Last: N/A
Push Attempts:             1
Push Succeeded:            1
Push Succeeded Last:      Fri Jul 28 15:32:16 2006
```

```
vSwitch:                   system
State:                      enabled
Sync Attempts:            0
Sync Succeeded:           0
Sync Succeeded Last:     N/A
Rcv Sync Succeeded:      0
Rcv Sync Succeeded Last: N/A
Push Attempts:            1
Push Succeeded:           1
Push Succeeded Last:     Fri Jul 28 15:32:16 2006

vSwitch:                   test
State:                      enabled
Sync Attempts:            0
Sync Succeeded:           0
Sync Succeeded Last:     N/A
Rcv Sync Succeeded:      0
Rcv Sync Succeeded Last: N/A
Push Attempts:            0
Push Succeeded:           0
Push Succeeded Last:     N/A
```

7. **Create a new virtual switch named scale3, then wait a few moments for the prompt to appear.**

```
sun(config)# vswitch scale3
Create new vSwitch "scale3"? (y or n): y
```

8. **Issue the following command and verify that the Configuration Synchronization State is enabled.**

```
sun(config-vSwitch-scale3)# show redundancy configSync

     ConfigSyncState:           enabled

     Sync Attempts:              0

     Sync Succeeded:             0

     Sync Succeeded Last:

     Rcv Sync Succeeded:        0

     Rcv Sync Succeeded Last:

     Push Attempts:              0

     Push Succeeded:             0

     Push Succeeded Last:
```

9. **Issue the following command.**

```
sun(config-vSwitch-scale3)# loadBalance
```

10. **Define a host for the vSwitch you just created.**

```
sun(config-vSwitch-scale3)# host scale3 1.2.3.4
```

11. **Verify the host was properly configured.**

```
sun(config-vSwitch-scale3)# show host
Name                   IP Address      Admin State  Description  vRouter
-------------------------- --------------- ------------ ------------ ----------
scale3                 1.2.3.4         enabled      N/A          scale3: default
```

12. **Create a vSwitch named scale3 on the Target switch.**

Note – Make certain you are using the CLI for the Target switch when you create this vSwitch.

```
sun(config)# vSwitch scale3

Create new vSwitch "scale3"? (y or n): y
```

13. **Push the host configuration from the Source switch to the Target switch.**

Note – Make certain you are using the CLI for the Source switch when you issue the `push` command.

```
sun(config-vSwitch-scale3 loadBalance)# snycCfg push
vSwitchName scale3
```

14. **Using the CLI for the Target switch, access the vSwitch that was just created.**

Note – Make certain you are using the CLI for the Target switch when you issue this command.

```
sun(config-vSwitch-scale3)# loadBalance
```

15. **Verify that the host on the Target switch was configured properly.**

```
sun(config-vSwitch-scale3 loadBalance)# show host
Name                IP Address     Admin State  Description  vRouter
-------------------------- ---------------- ------------ ------------ ----------
scale3              1.2.3.4        enabled      N/A          scale3: default
```

Establishing Configuration Synchronization: Using the snycCfg Command

At anytime, you can issue a sync command, synchronizing the load balance parameters from the Source switch to the Target switch. This command clears the load balance configuration on the Target switch and replaces it with the load balance configuration on the Source switch.

The syncCfg command can be used to synchronize several user defined vSwitches at one time. It is also useful when synchronizing a load balance configuration that was developed while troubleshooting, as a result of trial and error, or whenever it would require numerous steps to issue various commands using the command line interface.

Index