Important Information

Latest Software

We recommend that you install the most recent software release to stay up-to-date with the latest functional improvements, stability fixes, security enhancements and protection against new and evolving attacks.

Latest Documentation

The latest version of this document is at: http://supportcontent.checkpoint.com/documentation_download?ID=22932

To learn more, visit the Check Point Support Center http://supportcenter.checkpoint.com.

For more about this release, see the R76 home page http://supportcontent.checkpoint.com/solutions?id=sk105938.

Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>07 August 2016</td>
<td>• Updated Dynamic Routing [on page 30]</td>
</tr>
<tr>
<td></td>
<td>• Added IPv6 support</td>
</tr>
<tr>
<td></td>
<td>• Corrected: 10 CoreXL instances are supported. Not 8 [&quot;Configuring CoreXL on Virtual Systems&quot; on page 55].</td>
</tr>
<tr>
<td>7 May 2013</td>
<td>• Added link to Application Control and URL Filtering Administration Guide [&quot;Using Application Control and URL Filtering with VSX&quot; on page 68].</td>
</tr>
<tr>
<td></td>
<td>• Added link to Anti-Bot and Anti-Virus Administration Guide [&quot;Using Anti-Bot and Anti-Virus with VSX&quot; on page 69]</td>
</tr>
<tr>
<td></td>
<td>• Changed description of vsx_util change_mgmt_ip [&quot;change_mgmt_ip&quot; on page 177]</td>
</tr>
<tr>
<td></td>
<td>• Changed description of vsx_util change_mgmt_subnet [&quot;change_mgmt_subnet&quot; on page 179]</td>
</tr>
<tr>
<td>30 January 2013</td>
<td>First release of this document</td>
</tr>
</tbody>
</table>

Feedback

Check Point is engaged in a continuous effort to improve its documentation.

Please help us by sending your comments
mailto:cp_techpub_feedback@checkpoint.com?subject=Feedback on Check Point VSX R76 Administration Guide.
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Important Information</td>
<td>3</td>
</tr>
<tr>
<td>Introduction to VSX</td>
<td>10</td>
</tr>
<tr>
<td>VSX Overview</td>
<td>10</td>
</tr>
<tr>
<td>VSX Glossary</td>
<td>11</td>
</tr>
<tr>
<td>How VSX Works</td>
<td>11</td>
</tr>
<tr>
<td>Physical Network Topology</td>
<td>12</td>
</tr>
<tr>
<td>VSX Virtual Network Topology</td>
<td>12</td>
</tr>
<tr>
<td>SmartDashboard Toolbar</td>
<td>13</td>
</tr>
<tr>
<td>VSX Architecture and Concepts</td>
<td>14</td>
</tr>
<tr>
<td>The VSX Gateway</td>
<td>14</td>
</tr>
<tr>
<td>Management Server Connections</td>
<td>14</td>
</tr>
<tr>
<td>Management Interface</td>
<td>16</td>
</tr>
<tr>
<td>Virtual Devices</td>
<td>17</td>
</tr>
<tr>
<td>Virtual System</td>
<td>17</td>
</tr>
<tr>
<td>Virtual System in Bridge Mode</td>
<td>17</td>
</tr>
<tr>
<td>Virtual Routers</td>
<td>18</td>
</tr>
<tr>
<td>Virtual Switches</td>
<td>19</td>
</tr>
<tr>
<td>Interfaces</td>
<td>19</td>
</tr>
<tr>
<td>VSX Management Overview</td>
<td>22</td>
</tr>
<tr>
<td>Security Management Server Model</td>
<td>22</td>
</tr>
<tr>
<td>Multi-Domain Security Management Model</td>
<td>22</td>
</tr>
<tr>
<td>Management Model Comparison</td>
<td>23</td>
</tr>
<tr>
<td>Management Server Communication - SIC</td>
<td>24</td>
</tr>
<tr>
<td>VSX Traffic Flow</td>
<td>24</td>
</tr>
<tr>
<td>Overview</td>
<td>24</td>
</tr>
<tr>
<td>Context Determination</td>
<td>24</td>
</tr>
<tr>
<td>Security Enforcement</td>
<td>27</td>
</tr>
<tr>
<td>Forwarding to Destination</td>
<td>27</td>
</tr>
<tr>
<td>VSX Routing Concepts</td>
<td>27</td>
</tr>
<tr>
<td>Routing Overview</td>
<td>27</td>
</tr>
<tr>
<td>Routing Between Virtual Systems</td>
<td>27</td>
</tr>
<tr>
<td>Source-Based Routing</td>
<td>30</td>
</tr>
<tr>
<td>NAT</td>
<td>30</td>
</tr>
<tr>
<td>Dynamic Routing</td>
<td>30</td>
</tr>
<tr>
<td>VSX Clusters</td>
<td>31</td>
</tr>
<tr>
<td>High Availability</td>
<td>31</td>
</tr>
<tr>
<td>Virtual System Load Sharing (VSLS)</td>
<td>31</td>
</tr>
<tr>
<td>Configuring VSX</td>
<td>33</td>
</tr>
<tr>
<td>Overview</td>
<td>33</td>
</tr>
<tr>
<td>Rules &amp; Security Policies</td>
<td>34</td>
</tr>
<tr>
<td>Creating VSX Gateways</td>
<td>34</td>
</tr>
<tr>
<td>Creating a New VSX Gateway</td>
<td>34</td>
</tr>
<tr>
<td>Converting Gateways to VSX Gateways</td>
<td>37</td>
</tr>
<tr>
<td>Working with VSX Gateways</td>
<td>39</td>
</tr>
<tr>
<td>Changing VSX Gateway Definitions</td>
<td>39</td>
</tr>
<tr>
<td>Deleting a VSX Gateway</td>
<td>42</td>
</tr>
<tr>
<td>VSX Gateway Recovery</td>
<td>42</td>
</tr>
</tbody>
</table>
add_member_reconf...................................................................................................175
change_interfaces.......................................................................................................175
change_mgmt_ip.........................................................................................................177
change_mgmt_private_net..........................................................................................177
change_interfaces.......................................................................................................177
change_mgmt_subnet.................................................................................................179
convert_cluster ...........................................................................................................179
reconfigure..................................................................................................................179
remove_member.........................................................................................................180
show_interfaces..........................................................................................................180
Upgrade.......................................................................................................................181
view_vs_conf...............................................................................................................181
vsls..............................................................................................................................183

The cphaprojb Command .......................................................................................184
Command Options ....................................................................................................184
Introduction to VSX

In This Section:

- VSX Overview
- VSX Glossary
- How VSX Works
- SmartDashboard Toolbar

VSX (Virtual System Extension) is a security and VPN solution for large-scale environments based on the proven security of Check Point Security Gateway. VSX provides comprehensive protection for multiple networks or VLANs within complex infrastructures. It securely connects them to shared resources such as the Internet and/or a DMZ, and allows them to safely interact with each other. VSX is supported by IPS™ Services, which provide up-to-date preemptive security.

VSX incorporates the same patented Stateful Inspection and Software Blades technology used in the Check Point Security Gateway product line. Administrators manage VSX using a Security Management Server or a Multi-Domain Server, delivering a unified management architecture that supports enterprises and service providers.

A VSX Gateway contains a complete set of virtual devices that function as physical network components, such as Security Gateway, routers, switches, interfaces, and even network cables. Centrally managed, and incorporating key network resources internally, VSX lets businesses deploy comprehensive firewall and VPN functionality, while reducing hardware investment and improving efficiency.

VSX Overview

VSX (Virtual System Extension) is a security and VPN solution for large-scale environments based on the proven security of Check Point Security Gateway. VSX provides comprehensive protection for multiple networks or VLANs within complex infrastructures. It securely connects them to shared resources such as the Internet and/or a DMZ, and allows them to safely interact with each other. VSX is supported by IPS™ Services, which provide up-to-date preemptive security.

VSX incorporates the same patented Stateful Inspection and Software Blades technology used in the Check Point Security Gateway product line. Administrators manage VSX using a Security Management Server or a Multi-Domain Server, delivering a unified management architecture that supports enterprises and service providers.

A VSX Gateway contains a complete set of virtual devices that function as physical network components, such as Security Gateway, routers, switches, interfaces, and even network cables. Centrally managed, and incorporating key network resources internally, VSX lets businesses deploy comprehensive firewall and VPN functionality, while reducing hardware investment and improving efficiency.
VSX Glossary

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>VSX</td>
<td>Virtual System Extension - Check Point virtual networking solution, hosted on a single computer or cluster containing virtual abstractions of Check Point Security Gateways and other network devices. These virtual devices provide the same functionality as their physical counterparts.</td>
</tr>
<tr>
<td>VSX Gateway</td>
<td>Physical server that hosts VSX virtual networks, including all virtual devices that provide the functionality of physical network devices.</td>
</tr>
<tr>
<td>Management Server</td>
<td>The Security Management Server or a Multi-Domain Security Management used by administrators to manage the VSX virtual network and its security policies.</td>
</tr>
<tr>
<td>virtual device</td>
<td>Generic term for any VSX virtual network component.</td>
</tr>
<tr>
<td>Virtual System</td>
<td>A virtual device that provides the functionality of a physical Security Gateway with all supported Software Blades.</td>
</tr>
<tr>
<td>Virtual System in the Bridge Mode</td>
<td>A Virtual System that implements native layer-2 bridging instead of IP routing, thereby enabling deployment of Virtual Systems in an existing topology without reconfiguring the IP routing scheme.</td>
</tr>
<tr>
<td>Virtual Switch</td>
<td>A virtual device that provides the functionality of a physical switch in a VSX deployment.</td>
</tr>
<tr>
<td>Virtual Router</td>
<td>A virtual device that provides the functionality of a physical router in a VSX deployment.</td>
</tr>
<tr>
<td>Warp Link (wrp)</td>
<td>A virtual interface that is created automatically in a VSX topology.</td>
</tr>
</tbody>
</table>

How VSX Works

Each Virtual System works as a Security Gateway, typically protecting a specified network. When packets arrive at the VSX Gateway, it sends traffic to the Virtual System protecting the destination network. The Virtual System inspects all traffic and allows or rejects it according to rules defined in the security policy.

In order to better understand how virtual networks work, it is important to compare physical network environments with their virtual (VSX) counterparts. While physical networks consist of many hardware components, VSX virtual networks reside on a single configurable VSX Gateway or cluster that defines and protects multiple independent networks, together with their virtual components.
Physical Network Topology
The figure below shows a typical deployment with four physical Security Gateways, each protecting a separate network. Each Security Gateway is a separate, physical machine that is hard-wired to the perimeter router and its corresponding network.

VSX Virtual Network Topology
The example shows how a single VSX Gateway with four Virtual Systems protects all four networks.

Each Virtual System in a VSX environment works as an individual Security Gateway, providing the same security and networking functionality as a physical gateway. This example also shows:

- Four Virtual Systems, each handling packet traffic to and from discrete networks.
- One Virtual Switch providing connectivity for all the Virtual Systems to the Internet router.
• “Virtual” interfaces and network cables (known as **Warp Links**) providing point-to-point connections between the Virtual Systems and the Virtual Switch.

**SmartDashboard Toolbar**

You can use the SmartDashboard toolbar to do these actions:

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Open the SmartDashboard menu. When instructed to select menu options, click this button to show the menu." /></td>
<td>Open the SmartDashboard menu. When instructed to select menu options, click this button to show the menu. For example, if you are instructed to select <strong>Manage &gt; Users and Administrators</strong>, click this button to open the Manage menu and then select the <strong>Users and Administrators</strong> option.</td>
</tr>
<tr>
<td><img src="image" alt="Save current policy and all system objects." /></td>
<td>Save current policy and all system objects.</td>
</tr>
<tr>
<td><img src="image" alt="Open a policy package, which is a collection of Policies saved together with the same name." /></td>
<td>Open a policy package, which is a collection of Policies saved together with the same name.</td>
</tr>
<tr>
<td><img src="image" alt="Open the Database Revision Control window." /></td>
<td>Open the Database Revision Control window.</td>
</tr>
<tr>
<td><img src="image" alt="Change global properties." /></td>
<td>Change global properties.</td>
</tr>
<tr>
<td><img src="image" alt="Verify Rule Base consistency." /></td>
<td>Verify Rule Base consistency.</td>
</tr>
<tr>
<td><img src="image" alt="Install the policy on Security Gateways or VSX Gateways." /></td>
<td>Install the policy on Security Gateways or VSX Gateways.</td>
</tr>
<tr>
<td><img src="image" alt="Open SmartConsole." /></td>
<td>Open SmartConsole.</td>
</tr>
</tbody>
</table>
VSX Architecture and Concepts

In This Section:
- The VSX Gateway ........................................................................................................... 14
- Virtual Devices .............................................................................................................. 17
- VSX Management Overview .......................................................................................... 22
- VSX Traffic Flow ............................................................................................................ 24
- VSX Routing Concepts .................................................................................................. 27
- VSX Clusters .................................................................................................................. 30

A VSX Gateway is a physical machine that hosts virtual “networks”, consisting of virtual devices that provide the functionality of their physical network counterparts such as: Security Gateways, routers and switches.

A VSX Gateway performs the following tasks:
- Communicates with the management server to handle provisioning and configuration for all virtual devices
- Manages state synchronization for High Availability and for Load Sharing in cluster deployments.

The VSX Gateway

A VSX Gateway is a physical machine that hosts virtual “networks”, consisting of virtual devices that provide the functionality of their physical network counterparts such as: Security Gateways, routers and switches.

A VSX Gateway performs the following tasks:
- Communicates with the management server to handle provisioning and configuration for all virtual devices
- Manages state synchronization for High Availability and for Load Sharing in cluster deployments.

Management Server Connections

A management server (Security Management Server or Multi-Domain Server) connects to the VSX Gateway and provides provisioning and configuration services for virtual devices located on the VSX Gateway. You can connect the management server to the VSX Gateway using one of the following scenarios.
- Local Connection: The management server connects directly to the VSX Gateway using a dedicated management interface.
- Remote Connection: The management server connects remotely from an external or internal network by means of a router connected to a management interface. This method ensures segregation of management traffic from all other traffic.
Local Management Connection

When using a local management server (Security Management Server or Multi-Domain Security Management), all management traffic is handled by a dedicated management interface [DMI] that connects the management server with the VSX Gateway. The dedicated management interface IP address can be either private or public.

Remote Management Connection

When using a remote management server (Security Management Server or Multi-Domain Security Management), management traffic travels via an internal or external network to a VSX Gateway to the management interface. This architecture segregates management traffic from all other traffic passing through the VSX Gateway.
Check Point recommends that remote management connections use a dedicated management interface (DMI) that connects directly to a router or switch that leads to the external network or the Internet.

You can choose to use a non-dedicated management interface by connecting a Virtual Router or Virtual Switch to the management interface.

When management traffic passes through a Virtual Router or Virtual Switch, you must ensure that the associated Warp Link IP address originates from the remote network. Furthermore, if the remote management connection arrives via the Internet, you must assign a routable, public IP address.

**Management Interface**

A VSX deployment can be managed using one of the following interface schemes:

- **Dedicated Management Interface (DMI):** Uses a separate interface that is restricted to management traffic, such as provisioning, logging and monitoring
- **Non-Dedicated Management Interface:** Uses a shared internal or external interface that also carries routine user traffic

**Dedicated Management Interface (DMI)**

Check Point recommends that you use a DMI for management to segregate management traffic from routine “production” traffic enhanced performance, especially for end users.

**Non-Dedicated Management Interface**

When configuring a non-DMI deployment, you can define remote management connections only via a Virtual Switch or Virtual Router. Remote management connects via a Virtual System are not supported.
When using non-DMI for the following reasons:

- Provisioning and logging may degrade user performance.
- Non-DMI is irreversible - you cannot change a non-DMI gateway to DMI.

**Virtual Devices**

This section describes virtual network components and their characteristics.

**Virtual System**

A Virtual System is a virtual security and routing domain that provides the functionality of a Security Gateway with full Firewall and VPN facilities. Multiple Virtual Systems can run concurrently on a single VSX Gateway.

**Virtual System Autonomy**

Each Virtual System functions as an independent entity, much in the same way as each Security Gateway is independent from other Gateways. Each Virtual System maintains its own Software Blades, interfaces, IP addresses, routing table, ARP table and dynamic routing configuration. In addition, each Virtual System maintains its own:

- **Software Blades**: You can configure different Software Blades for each Virtual System to give the applicable protection to the networks.
- **State Tables**: Each Virtual System contains its own kernel tables containing configuration and runtime data, such as, active connections, IPSec tunnel information, and so on.
- **Security and VPN policies**: Each Virtual System enforces its own security and VPN Policies (including INSPECT code). Policies are retrieved from the management server and stored separately on the local disk and in the kernel. In a Multi-Domain Security Management environment, each Domain database is maintained separately on the management server as well as on the VSX Gateway.
- **Configuration Parameters**: Each Virtual System maintains its own configuration, such as IPS settings, TCP/UDP time-outs, and so on. Different Virtual Systems can run in layer-2 or layer-3 mode and co-exist on the same VSX Gateway.
- **Logging Configuration**: Each Virtual System maintains its own logs and performs logging according to its own rules and configuration.

**Virtual System in Bridge Mode**

A Virtual System in bridge mode implements native layer-2 bridging instead of IP routing. This lets you easily and transparently deploy a Virtual System in an existing network topology without reconfiguring the existing IP routing scheme.
A typical bridge mode scenario incorporates an 802.1q compatible VLAN switch on either side of the VSX Gateway. The Virtual System interfaces do not require IP addresses and it remains transparent to the existing IP network.

A Virtual System in the bridge mode:
- Has the same security capabilities as a Virtual System, except for VPN and NAT
- Simplifies virtual network management
- Does not segment an existing virtual network
- Requires manual topology configuration in order to enforce Anti-Spoofing

**Virtual Routers**

A Virtual Router is an independent routing domain within a VSX Gateway that performs the functionality of physical routers. Virtual Routers are useful for connecting multiple Virtual Systems to a shared interface, such as the interface leading to the Internet, and for routing traffic from one Virtual System to another. Virtual Routers support dynamic routing.

Virtual Routers perform the following routing functions:
- Packets arriving at the VSX Gateway through a shared interface to the designated Virtual System based on the source or destination IP address.
- Traffic arriving from Virtual Systems directed to a shared interface or to other Virtual Systems.
- Traffic to and from shared network resources such as a DMZ.

As with physical routers, each Virtual Router maintains a routing table with a list of route entries describing known networks and directions on how to reach them. Depending on the deployment requirements, multiple Virtual Routers can be configured.

To protect themselves, Virtual Routers inspect all traffic destined to, or emanating from themselves (for example, an ICMP ping to the Virtual Router IP address) based on the security
policy. Traffic that is not sent to, or coming from the Virtual Router is not inspected by the Virtual Router policy and is sent to its destination.

**Virtual Switches**

By providing layer-2 connectivity, a **Virtual Switch** connects Virtual Systems and facilitates sharing a common physical interface without segmenting the existing IP network. As with a physical switch, each Virtual Switch maintains a forwarding table with a list of MAC addresses and their associated ports.

In contrast to a Virtual Router, when sharing a physical interface via a Virtual Switch there is no need:

- To allocate an additional subnet for IP addresses of Virtual Systems connected to the switch.
- To manually configure the routing on the routers adjacent to the shared interface.

You can create multiple Virtual Switches in a virtual network topology.

**Note** - When sharing a physical interface via a Virtual Switch, the IP addresses for Virtual Systems connected to a Virtual Switch should be allocated from the same subnet as the shared interface.

If the only function the Virtual Switch performs is to connect Virtual Systems, then the Virtual Switch can be defined without interfaces (unless Virtual System Load Sharing is enabled).

**Interfaces**

This section describes the various types of interfaces and how they are used in a VSX configuration. The principal interface types are:

- Physical Interface
- VLAN interface
- Warp Link (including unnumbered interfaces)
The following figure presents a simple example that illustrates how the various interface types are used in a VSX environment.

**Notes:**
- Warp Links connect the Virtual Switch to each Virtual System.
- A Physical Interface connects the Virtual Switch to an external router leading to the Internet.
- VLAN Interfaces connect the Virtual Systems to the VLAN Switch, via a VLAN trunk.
- The VLAN switch connects to the protected networks.

**Physical Interfaces**
Physical interfaces connect a VSX Gateway to internal and external networks, as well as to the management server. There are different types of physical interfaces (four types for a VSX Cluster) used in a VSX Gateway:

- **Dedicated Management Interface:** Connects the VSX Gateway to the management server when it is locally managed. If the VSX Gateway is remotely managed, then the management connection arrives via the external or internal interface.
- **External interface:** Connects the VSX Gateway to the Internet or other untrusted networks.
- **Internal Interface:** Connects the VSX Gateway to a protected network.
- **Synchronization Interface:** Connects one VSX Gateway member to other members for state synchronization in a VSX clustering deployment.

Additional physical interfaces can be installed and attached to any virtual device as required. A VSX Gateway can theoretically contain as many physical interfaces as permitted by gateway hardware and memory constraints.
**VLAN Interfaces**

Virtual Systems typically connect to protected VLAN networks using IEEE 802.1q compliant VLAN Interfaces. The networks are connected to ports on an 802.1q-compliant switch that trunks all traffic via a single physical interface to the VSX Gateway.

VSX uses VLAN tags to direct the Ethernet frames to the specific Virtual System handling each network. VSX assigns a virtual VLAN interface to each VLAN tag on a specific physical interface. For Example: VLAN tag 100 on eth3 will be assigned a virtual interface named eth3.100.

**Warp Links**

A Warp Link is a virtual point-to-point connection between a Virtual System and a Virtual Router or Virtual Switch. Each side of a Warp Link represents is a virtual interface with the appropriate virtual device.

R76 VSX automatically assigns a name to each virtual interface when administrators create the link. Warp Interfaces on the Virtual System side are assigned the prefix \textit{ wrp} and those on the Virtual Router / Virtual Switch side are assigned the prefix \textit{ wrpj}. In both cases, VSX appends a unique number to the prefix to form the interface name.

When connected to a Virtual Switch, VSX also assigns a unique MAC address to each Warp Link.

**Unnumbered Interfaces**

VSX lets you reduce the number of IP addresses required for a VSX network deployment when using one or more Virtual Routers. A Warp Link connected to a Virtual Router can “borrow” an existing IP address from another interface, instead of assigning a dedicated address to the interface leading to a Virtual Router. This capability is known as an \textit{ Unnumbered Interface}.

The previous figure illustrates a topology using unnumbered interfaces. In this example, the external interfaces for each Virtual System are unnumbered and borrow the IP address of the internal interfaces. Unnumbered interfaces act as the next hop from the Virtual Router.

**Unnumbered Interface Limitations**

The following limitations apply to Unnumbered Interfaces:

- Unnumbered interfaces must connect to a Virtual Router.
VSX Management Overview

VSX supports two Check Point management models: Security Management Server and Multi-Domain Security Management. Both models provide central configuration, management and monitoring for multiple VSX Gateways and Virtual Systems. The choice of management model depends on several factors, including:

- The scale of the current deployment and anticipated expansion
- Administrative requirements
- Physical and operational requirements
- Licensing restrictions

You can use either management model to manage a “physical” Security Gateway together with a VSX Gateway and Virtual Systems. You can also manage VPN communities and remote connections with either model.

**Note** - According to the Check Point EULA (End User License Agreement), a Security Gateway can only manage security policies for Virtual Systems belonging to a single legal entity. In order to manage Virtual Systems belonging to multiple legal entities, you need to deploy a Multi-Domain Security Management solution with a separate Domain Management Server for each legal entity. For more information regarding Licensing, refer to your Check Point Reseller.

Security Management Server Model

The Security Management Server model is appropriate for enterprise deployments containing many Virtual Systems. In this model, SmartDashboard connects to the VSX Gateway, which contains the Virtual Systems, and directly manages each Virtual System.

There is a single management domain with one object database to manage virtual devices as well as other physical devices. Only one administrator at a time can use SmartDashboard to provision Virtual Switches, and configure security policies.

Multi-Domain Security Management Model

Using the Multi-Domain Security Management model, administrators centrally manage multiple independent networks, typically belonging to different Domains, divisions or branches. The Multi-Domain Server is the central management node that controls the network and security policy databases for each of these networks.
Each Domain network is managed by a **Domain Management Server**, which provides the full functionality of a Security Management Server and can host multiple Virtual Systems, virtual and physical devices. The server that manages the VSX Gateway is the **Main Domain Management Server**. A VSX Gateway can host Virtual Systems that are managed by different Domain Management Servers.

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SmartDomain Manager</td>
</tr>
<tr>
<td>2</td>
<td>Multi-Domain Server</td>
</tr>
<tr>
<td>3</td>
<td>SmartDashboard</td>
</tr>
<tr>
<td>4</td>
<td>Domain Management Server</td>
</tr>
<tr>
<td>5</td>
<td>Main Domain Management Server</td>
</tr>
<tr>
<td>6</td>
<td>VSX Gateway</td>
</tr>
<tr>
<td>7</td>
<td>Virtual Systems in Domain Management Servers</td>
</tr>
</tbody>
</table>

Using the **SmartDomain Manager**, you provision and configure Domains and Domain Management Servers. Each Domain Management Server uses its own SmartDashboard instance to provision and configure its Virtual Systems, virtual devices, and security policies.

**Management Model Comparison**

The following table summarizes the capabilities and differences between the two management models. The capacity figures shown for Multi-Domain Security Management represent estimated, practical limits that will sustain acceptable performance levels under normal conditions. Actual performance is dependent on many factors, including deployed hardware, network topology, traffic load and security requirements.
Management Server Communication - SIC

All communication between the management server and the VSX Gateway is accomplished by means of Secure Internal Communication (SIC), a certificate based channel that authenticates communication between Check Point components. The management server uses SIC for provisioning virtual devices, policy installation, logging, and status monitoring.

SIC trust is initially established using a one-time password during configuration of the VSX Gateway or VSX cluster members. For Multi-Domain Security Management deployments, SIC trust is established between the Domain Management Server associated with the VSX Gateway or VSX cluster (Main Domain Management Server).

The virtual devices establish trust in a different manner than their physical counterparts. When creating a virtual device, VSX automatically establishes SIC trust using the secure communication channel defined between the management server and the VSX Gateway. The VSX Gateway uses its management interface for Secure Internal Communication between the management server and all virtual devices.

VSX Traffic Flow

Overview

A VSX Gateway processes traffic according to the following steps:

- Context determination
- Security enforcement
- Forwarding to destination

Context Determination

VSX incorporates **VRF** (Virtual Routing and Forwarding) technology that allows creation of multiple, independent routing domains on a single VSX Gateway or VSX cluster. The independence of these routing domains makes possible the use of virtual devices with overlapping IP addresses. Each routing domain is known as a **context**.

When traffic arrives at a VSX Gateway, a process known as **Context Determination** directs traffic to the appropriate Virtual System, Virtual Router or Virtual Switch. The context determination process depends on the virtual network topology and the connectivity of the virtual devices.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Security Management Server</th>
<th>Multi-Domain Security Management (Practical Limit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management Domains</td>
<td>1</td>
<td>250</td>
</tr>
<tr>
<td>Concurrent Administrators</td>
<td>1</td>
<td>250</td>
</tr>
<tr>
<td>Object Databases</td>
<td>1</td>
<td>250</td>
</tr>
<tr>
<td>Policies</td>
<td>250</td>
<td>250</td>
</tr>
<tr>
<td>Certificate Authorities</td>
<td>1</td>
<td>250</td>
</tr>
<tr>
<td>Virtual Systems</td>
<td>25 (recommended)</td>
<td>250</td>
</tr>
</tbody>
</table>
The basic Virtual System connection scenarios are:

- Virtual System directly connected to a physical or VLAN interface
- Virtual System connected via a Virtual Switch
- Virtual System connected via a Virtual Router

**Direct Connection to a Physical Interface**

When traffic arrives at an interface (either physical or VLAN) that directly connects to a Virtual System, the connection itself determines the context and traffic passes directly to the appropriate Virtual System via that interface. This diagram shows traffic from a physical VLAN switch that is sent to an interface on the VSX Gateway.

VSX automatically directs traffic arriving via VLAN Interface eth1.200 to Virtual System 2 according to the context defined by the VLAN ID.
**Connection via a Virtual Switch**

Traffic arriving via a Virtual Switch passes to the appropriate Virtual System based on the destination MAC address, as defined in the Virtual Switch forwarding table. Traffic arrives at the Virtual System via the Warp Link associated with the designated MAC address.

If the destination MAC address does not exist in the Virtual Switch forwarding table, the traffic is broadcast over all defined Warp Links. The Virtual Switch scenario is common for inbound traffic from external networks or the Internet.
**Connection via a Virtual Router**

Traffic arriving via a Virtual Router passes to the appropriate Virtual System based on entries in the Virtual Router routing table. Routing may be destination-based, source-based or both. Traffic arrives to the designated Virtual System via its Warp Link.

![VSX Architecture Diagram](image)

**Security Enforcement**

Since each Virtual System functions as an independent Security Gateway, it maintains its own, unique security policy to protect the network behind it. The designated Virtual System inspects all traffic and allows or blocks it based on the rules contained in the security policy.

**Forwarding to Destination**

Each Virtual System maintains its own unique configuration and rules for processing and forwarding traffic to its final destination. This configuration also includes definitions and rules for NAT, VPN, and other advanced features.

**VSX Routing Concepts**

**Routing Overview**

The traffic routing features in VSX network topologies are analogous to those available for physical networks. This section discusses several routing features and strategies as they apply to a VSX environment.

**Routing Between Virtual Systems**

Virtual Routers and Virtual Switches can be used to send traffic between networks located behind Virtual Systems, much in the same way as their physical counterparts.
The figure below shows an example of how Virtual Systems, connected to a Virtual Switch and a physical VLAN switch, communicate with each other. In this example, a host in VLAN 100 sends data to a server located in VLAN 200.

1. Traffic from the VLAN 100 host arrives at the VLAN switch, which inserts a VLAN tag and sends it to the VSX Gateway by way of a VLAN trunk.
2. Based on its VLAN tag, the VSX Gateway assigns the traffic to the Virtual System named VS1.
3. VS1 inspects the traffic according to its security policy and sends the traffic on to the Virtual Switch. VS1 “knows” to send the traffic to VS2 by way of the Virtual Switch, based on its routing configuration.
4. VS2 inspects the traffic according to its security policy, inserts a VLAN tag, and sends it back to the VLAN switch.
5. The VLAN switch sends the traffic to the server located on VLAN 200.

**Route Propagation**

When a Virtual System is connected to a Virtual Router or to a Virtual Switch, you can choose to propagate its routing information to adjacent virtual devices. This feature enables network nodes located behind neighboring Virtual Systems to communicate without the need for manual configuration.

Route propagation works by automatically updating virtual device routing tables with routes leading to the appropriate Virtual Systems.

**Route Propagation using a Virtual Router**

When Virtual Systems are connected to a Virtual Router, VSX propagates routes by automatically adding entries to the routing table contained in the Virtual Router. Each entry contains a route pointing to the destination subnet using the Virtual System router-side Warp Interface (\texttt{wrpj}) as the next hop.
Route Propagation using a Virtual Switch

When Virtual Systems are connected to a Virtual Switch, VSX propagates routes by automatically adding entries to the routing table in each Virtual System. Each entry contains a route pointing to the destination subnet using the Virtual System Warp Interface (wrp) IP address.

Overlapping IP Address Space

VSX facilitates connectivity when multiple network segments share the same IP address range (IP address space). This scenario occurs when a single VSX Gateway protects several independent networks that assign IP addresses to endpoints from the same pool of IP addresses. Thus, it is feasible that more than one endpoint in a VSX environment will have the identical IP address, provided that each is located behind different Virtual System.

Overlapping IP address space in VSX environments is possible because each Virtual System maintains its own unique state and routing tables. These tables can contain identical entries, but within different, segregated contexts. Virtual Systems use NAT to facilitate mapping internal IP addresses to one or more external IP addresses.

The below figure demonstrates how traffic passes from the Internet to an internal network with overlapping IP address ranges, using NAT at each Virtual System.

In this case, Network 1, Network 2, Network 3, and Network 4 all share the same network address pool, which might result in identical overlapping IP addresses. However, packets originating from or targeted to these networks are processed by their respective Virtual System using NAT to translate the original/overlapping addresses to unique routable addresses.

Additional Considerations for Virtual Switch Route Propagation

To update the topology map for each Virtual System, you still need to edit and save each Virtual System object that is connected to the Virtual Switch after enabling route propagation. You do not, however, need to manually define the topology, as this is done automatically.
Following the topology update, you must then re-install the security policy for the affected Virtual Systems. This procedure is necessary in order to ensure that the Anti-Spoofing and VPN features work properly.

**Source-Based Routing**

Source-based routing allows you to create routing definitions that take precedence over ordinary, destination-based, routing decisions. This lets you route packets according to their source IP address or a combination of their source IP address and destination IP address.

Source-based routing is useful in deployments where a single physical interface without VLAN tagging connects several protected Domain networks. All Virtual Systems are connected to an internal Virtual Router. The Virtual Router sends traffic to the applicable Virtual System based on the source IP address, as defined in source-based routing rules.

**Limitations**

- Source-based routing does not support overlapping IP addresses.
- Anti-Spoofing protection is not effective for packets that originate from a shared internal interface, because there is no physical or logical segregation of traffic. In this case, it is recommended that you configure Anti-Spoofing protection on the router itself.

**NAT**

Virtual Systems support Network Address Translation (NAT), much in the same manner as a physical firewall. When a Virtual System, using either Static or Hide NAT, connects to a Virtual Router, you must propagate the affected routes to the Virtual Router. To do so, you need to first define NAT addresses for Virtual Systems connected to a Virtual Router.

The NAT configuration section ("Virtual System - NAT > Advanced" on page 48) presents the configuration procedure for NAT on Virtual Machines.

**Dynamic Routing**

The virtual devices can communicate and distribute routes using dynamic routing. Each virtual device has its own routing daemon.

Virtual Systems support:

- OSPF
- RIP
- BGP
- PIM

Virtual Routers support:

- OSPF
VSX Clusters

A VSX cluster consists of two or more identical, interconnected VSX Gateways that ensure continuous data synchronization and transparent failover. Furthermore, Virtual System Load Sharing (VSLS) enhances throughput by distributing Virtual Systems, together with their traffic load, amongst multiple, redundant machines.

VSX supports the following cluster environments:

- Check Point ClusterXL
- Crossbeam X-Series Chassis

VSX supports the following **Bridge Mode** solutions for ClusterXL deployments:

- **STP Bridge Mode**: Provides path redundancy while preventing undesirable loops between redundant switches.
- **Active/Standby Bridge Mode**: Provides full path redundancy and loop prevention, while offering seamless support for Virtual System Load Sharing and overcomes many STP limitations.

The VSX Clusters chapter ("Introduction to VSX Clusters" on page 74) provides detailed conceptual information, while the Cluster Management chapter ("Working with VSX Clusters" on page 85) provides detailed configuration procedures, including instructions for enabling and using all VSX clustering features. For more about Check Point ClusterXL features and functionality see the *R76 ClusterXL Administration Guide*.

High Availability

VSX provides High Availability and transparent failover for VSX Gateways and/or for Virtual Systems. If the active VSX Gateway member fails, all sessions continue to run, securely and without interruption, on a standby cluster member. If an individual Virtual System fails, you can configure that Virtual System to fail over to a standby member while all other Virtual Systems continue to function on the active VSX Gateway member.

Users need not reconnect and re-authenticate, nor do they notice that an alternate machine has taken over. The Selective Sync feature allows you to selectively activate, delay or disable cluster member synchronization.

Virtual System Load Sharing (VSLS)

Load Sharing offers significant performance advantages while providing failover for individual Virtual Systems. Using multiple Gateways instead of a single gateway significantly increases performance for CPU intensive applications such as VPNs, Security servers, Policy servers, and Active Directory (LDAP).

By distributing Virtual System instances between different cluster members, the performance load is efficiently spread amongst the members. For example, active Virtual System 1 runs on member A, while active Virtual System 2 runs on member B. Standby and backup Virtual System instances are likewise distributed amongst members to maximize throughput, even in a failover scenario.
VSLS provides an excellent scalability solution, allowing administrators to add additional physical members to an existing VSLS cluster as traffic loads and performance requirements increase. VSLS is available only in a Check Point ClusterXL environment.
Configuring VSX

In This Section:

Overview ........................................................................................................................ 33
Creating VSX Gateways ............................................................................................... 34
Working with VSX Gateways ...................................................................................... 39
Working with Virtual Systems .................................................................................... 42
Working with Virtual Switches .................................................................................... 48
Creating a New Virtual Switch .................................................................................... 50
Working with Virtual Routers ..................................................................................... 50
CoreXL for Virtual Systems ......................................................................................... 54
Dynamic Routing for virtual devices ......................................................................... 56
Working with Interface Definitions .......................................................................... 56
Working with Authentication ...................................................................................... 59
Tracking Activity with SmartView Monitor ................................................................. 65
Client/Session Authentication ..................................................................................... 65
Working with Network Address Translation ............................................................... 67
Using Application Control and URL Filtering with VSX ........................................... 68
Using Anti-Bot and Anti-Virus with VSX ................................................................. 69
Licensing VSX ........................................................................................................... 69

This chapter shows you how to use SmartDashboard to provision, configure and manage virtual devices in a VSX environment.

If you define or configure VSX objects in a Multi-Domain Security Management deployment: open the SmartDashboard of the Domain Management Server that manages the virtual devices. The Multi-Domain Security Management chapter (“Using VSX with Multi-Domain Security Management” on page 71) explains these procedures.

To configure virtual devices, make sure that these preparations are ready:

- The management servers (Security Management Server or Multi-Domain Server) are configured and running.
- The GUI clients (SmartDashboard or SmartDomain Manager) are installed on the appropriate computers.

This chapter assumes that you are familiar with SmartDashboard and how to configure standard Security Gateway objects and security policies. Many virtual device and policy operations are the same as physical Security Gateways and these standard procedures are not in this Administration Guide.

Overview

This chapter explains how to use SmartDashboard provision, configure and manage virtual devices in a VSX environment.

If you define or configure VSX objects in a Multi-Domain Security Management deployment: open the SmartDashboard of the Domain Management Server that manages the virtual devices. The
Multi-Domain Security Management chapter ["Using VSX with Multi-Domain Security Management" on page 71] explains these procedures.

To configure virtual devices, make sure that these preparations are ready:

- The management servers (Security Management Server or Multi-Domain Server) are configured and running.
- The GUI clients (SmartDashboard or SmartDomain Manager) are installed on the appropriate computers.

This chapter assumes that you are familiar with SmartDashboard and how to configure standard Security Gateway objects and security policies. Many virtual device and policy operations are the same as physical Security Gateways and these standard procedures are not in this Administration Guide.

**Rules & Security Policies**

Defining and installing security policies on a VSX gateway or Virtual System is the same as on a physical Security Gateway. Therefore, these procedures are not presented in this Administration Guide.

**Important** - The Revision Control feature is not supported when the Security Management Server database contains VSX objects. You must not select the Create database version option in SmartDashboard when you install a policy.

**Creating VSX Gateways**

**Creating a New VSX Gateway**

This section explains how to create a new VSX Gateway using the VSX Gateway Wizard. After you complete the VSX Gateway Wizard, you can change the VSX Gateway definition from SmartDashboard. For example, you can add or delete interfaces, or configure existing interfaces to support VLANs.

To start the VSX Gateway wizard:

1. Open SmartDashboard.
   - If you are using Multi-Domain Security Management, open SmartDashboard from the Domain Management Server of the VSX Gateway.
2. From the Network Objects tree, right-click Check Point and select VSX > Gateway.
   - The General Properties page of the VSX Gateway Wizard opens.

**Defining VSX Gateway General Properties**

The General Properties page contains basic identification properties for VSX Gateways.

- **VSX Gateway Name**: Unique, alphanumeric for the VSX Gateway. The name cannot contain spaces or special characters except the underscore.
- **VSX Gateway IP Address**: Management interface IP address.
- **VSX Gateway Version**: Select the VSX version installed on the VSX Gateway from the drop-down list.
**Selecting Creation Templates**

The **Creation Templates** page lets you provision predefined, default topology and routing definitions to Virtual Systems. This makes sure Virtual Systems are consistent and makes the definition process faster. You always have the option to override the default creation template when you create or change a Virtual System.

The default Creation Templates are:

- **Shared Interface**: Virtual Systems share one external interface, but maintain separate internal interfaces.
- **Separate Interfaces**: Virtual Systems use their own separate internal and external interfaces. This template creates a Dedicated Management Interface (DMI) by default.

If the default templates are not appropriate, you can create a custom configuration:

- **Custom Configuration**: Define Virtual System, Virtual Router, Virtual Switch, and Interface configurations.

**Establishing SIC Trust**

Initialize Secure Internal Communication trust between the VSX Gateway and the management server. The gateway and server cannot communicate without Trust.

**Initializing SIC Trust**

When you create a VSX Gateway, you must enter an Activation Key. Enter and confirm the activation key and then click **Initialize**. If you enter the correct activation key, the Trust State changes to **Trust established**.

**Troubleshooting SIC Trust Initialization Problems**

If SIC trust was not successfully established, click **Check SIC Status** to see the reason for the failure. The most common issues are an incorrect activation key and connectivity problems between the management server and the VSX Gateway.

Troubleshooting to resolve SIC initialization problems:

- Re-enter and re-confirm the activation key.
- Verify that the IP address defined in **General Properties** is correct.
- Ping the management server to verify connectivity. Resolve connectivity issues.
- From the VSX Gateway command line, use the `cpconfig` utility to re-initialize SIC. After this process completes, click **Reset** in the wizard and then re-enter the activation key.


**Defining Physical Interfaces**

In the **VSX Gateway Interfaces** window, define physical interfaces as VLAN trunks. The page shows the interfaces currently defined on the VSX Gateway.

To define an interface as a VLAN trunk, select **VLAN Trunk** for the interface.
Virtual Network Device Configuration

If you chose the Custom Configuration option, the Virtual Network Device Configuration window opens. In this window, define a virtual device with an interface shared with the VSX Gateway. If you do not want to define a virtual device at this time, click Next to continue.

To define a virtual device with a shared interface:

1. Select Create a virtual device.
2. Select the Virtual Network Device type (Virtual Router or Virtual Switch).
3. Select the shared physical interface to define a non-DMI gateway.
   Do not select the management interface if you want to define a Dedicated Management Interface (DMI) gateway. If you do not define a shared virtual device, a DMI gateway is created by default.
   Important - This setting cannot be changed after you complete the VSX Gateway Wizard. If you define a non-DMI gateway, you cannot change it to a DMI gateway later.
4. Define the IP address and Net Mask for a Virtual Router.
   These options are not available for a Virtual Switch.
5. Optional: Define a Default Gateway for a Virtual Router (DMI only).

VSX Gateway Management

In the VSX Gateway Management window, define security policy rules that protect the VSX Gateway. This policy is installed automatically on the new VSX Gateway.

Note - This policy applies only to traffic destined for the VSX Gateway. Traffic destined for Virtual Systems, other virtual devices, external networks, and internal networks is not affected by this policy.

The security policy consists of predefined rules for these services:
• **UDP** - SNMP requests
• **TCP** - SSH traffic
• **ICMP** - Echo-request (ping)
• **TCP** - HTTPS traffic

**Completing the VSX Wizard**

Click **Next** to continue and then click **Finish** to complete the VSX Gateway wizard.

This may take several minutes to complete. A message shows successful or unsuccessful completion of the process.

If the process ends unsuccessfully, click **View Report** to see the error messages. See the Troubleshooting chapter ("VSX Diagnostics and Troubleshooting" on page 157).

**Configuring the Gateway Security Policy**

1. **Allow**: Select to pass traffic on the selected services. Clear this option to block traffic on this service. By default, all services are blocked.
   
   For example, to be able to ping the gateway from the management server, allow ICMP echo-request traffic.

2. **Source**: Click the arrow and select a **Source Object** from the list.
   
   The default value is *Any*. Click **New Source Object** to define a new source.

**Converting Gateways to VSX Gateways**

Use the VSX Gateway Conversion wizard in SmartDashboard to convert Gaia Security Gateways to VSX Gateways. You can convert one Security Gateway or all the members of a cluster to VSX. The settings of the Security Gateways are applied to the VSX Gateway (VS0). You can also use SmartDashboard to convert a VSX Gateway to a Security Gateway.

We recommend that you go to sk79260 http://supportcontent.checkpoint.com/solutions?id=sk79260, before you use the Conversion wizard. You can only convert Security Gateways or clusters that use the Gaia operating system.

**Note** - The Security Gateway loses connectivity during the conversion process.

**Converting a Security Gateway**

SmartDashboard converts a Security Gateway or cluster to VSX. You can only complete the Conversion Wizard if the features and settings of the Security Gateway or cluster are compatible with VSX.

When the **Conversion Process** window is shown, you cannot cancel or close the Conversion Wizard.

**To convert a Security Gateway:**

1. Open SmartDashboard.
2. In the **Network Objects** tree, right-click the Security Gateway or cluster and select **Convert to VSX**.
3. When the **Welcome to the VSX Conversion** window opens, click **Next** to continue.
4. In the **Compatibility Check** window, click **Next to continue**.
   The compatibility check makes sure that the Security Gateway or cluster is compatible with VSX.

5. In the **Security Management Server Interface Sharing** window, configure how interfaces are created for the new Virtual Systems and then click **Convert**.

6. After the conversion process completes, click **Finish**.
   The **Converting** window shows as the management database is updated.

   _Note_ - You cannot use SmartDashboard while the **Converting** window shows.

### Checking Compatibility

The VSX Gateway Conversion Wizard cannot convert a Security Gateway or cluster that uses Software Blades or other features that VSX does not support. The wizard automatically checks for common compatibility problems with the Security Gateway. We recommend that you go to sk79260 http://supportcontent.checkpoint.com/solutions?id=sk79260, to see a full list of limitations and compatibility problems.

If the Security Gateway is not compatible, the **Compatibility Check** window tells you the solution for each compatibility problem. Close the wizard, disable the unsupported features, and run the VSX Gateway Conversion Wizard again.

### Completing the Conversion

Complete the Security Gateway to VSX Gateway Conversion Wizard. When you complete the wizard, the management database is updated with the new VSX Gateway object.

To complete the Conversion Wizard:

Click **Finish**. The **Converting** window is shown as the management database is updated.

   _Note_ - You cannot use SmartDashboard while the **Converting** window is shown.

### Converting a VSX Gateway

SmartDashboard converts a VSX Gateway or cluster to a Security Gateway. You must remove all the Virtual Systems and other virtual devices from the VSX object before you can convert the VSX Gateway.

You cannot convert a VSX Gateway that uses a shared interface configuration to a Security Gateway.

To convert a VSX Gateway to a Security Gateway:

1. Remove all the virtual devices from the VSX object.
   From the **Network Objects** tree, right-click each virtual device object and select **Delete**.

2. Right-click the VSX Gateway or cluster and select **Convert to Gateway**.
   A confirmation window opens.

3. Click **Yes**.
   The VSX Gateway is converted to a Security Gateway.

   _Note_ - You cannot use SmartDashboard while the **Converting** window is shown.
Working with VSX Gateways

A VSX gateway is a physical machine that serves as a container for Virtual Systems and other virtual network components. This section has step-by-step procedures for creating and configuring standalone VSX gateways.

Changing VSX Gateway Definitions

After you create a VSX Gateway, you can modify the topology, other parameters, and advanced configurations in the **VSX Gateway Properties** window. To open this window, double-click on the VSX Gateway object in SmartDashboard. The **VSX Gateway Properties** window opens, showing the **General Properties** page.

**VSX Gateway - General Properties**

In the **General Properties** page, check and re-establish SIC trust, and activate Check Point products for this VSX Gateway.

You can change these properties:

- **Comment** - Free text description for the Object List and elsewhere.
- **Color** - Color of the object icon as it appears in the Object Tree.
- **Secure Internal Communication** - Check and re-establish SIC trust.
- **Check Point Products** - Select Check Point products for this gateway.

**Secure Internal Communication (SIC)**

You can test and reset SIC trust and also see the VSX Gateway Relative Distinguished Name.

To initialize SIC trust:

1. Open SmartDashboard.
2. From the **Network Objects** tree, right-click the VSX Gateway and select **Edit**.
   
   The **VSX Gateway Properties** window opens.
3. Click **Communication**.
   
   The **Trusted Communication** window opens.
4. Enter and confirm the SIC authentication password.
5. Click **Initialize**.

**Note** - If you cannot establish trust, click **Test SIC Status** to see the reason for the failure. The most common issues are an incorrect activation key and connectivity problems between the management server and the VSX Gateway.

To reset SIC trust with the VSX Gateway:

1. From the VSX Gateway CLI, use the `cpconfig` utility to re-initialize the SIC.
2. In the **Communication** window, click **Reset**.
3. Click **Yes** in the confirmation window.
4. Enter and confirm the SIC authentication password.
5. Click **Initialize**.
6. Install policy to VS0 only.
7. On each member, run: `cpstop`; `cpstart`

**Check Point Software Blades**

Select the Check Point Software Blades to install on this VSX Gateway from the list. The items you see are available for the product version and your license agreement.

**VSX Gateway - Creation Templates**

The **Creation Templates** page displays the creation template used to create the Virtual Systems for this VSX Gateway. You can change from the current creation template to the **Custom Configuration** template and change the shared physical interface if the Shared Interface template is active.

- Select **Custom Configuration** to change from the Shared Interfaces or Separate Interfaces templates. You cannot change back from the **Custom Configuration** template once you have completed the definition and saved it to the VSX Gateway.
- For a **Shared Interface** template, click **Settings** to change the shared interface.

**VSX Gateway - Physical Interfaces**

The **Physical Interfaces** page lets you add or delete a physical interface on the VSX Gateway, and to define a VLAN trunk.

- To add a new physical interface, click **Add** and enter the interface name in the appropriate field.
- To remove a physical interface, select the interface and click **Remove**.
- To define an interface as a VLAN trunk, select **VLAN Trunk** for the interface.

**VSX Gateway - Topology**

The **Topology** page contains definitions for interfaces and routes between interfaces and virtual devices.

**Interfaces**

The Interfaces section defines interfaces and links to devices. You can add new interfaces, and delete or modify existing interfaces.

**To add an interface:**

1. Click **New** and select one of these options:
   - **Regular** - Create a new interface
   - **Leads to Virtual Router**
   - **Leads to Virtual Switch**
   
   The **Interface Properties** window opens.
   
   Click **Actions > Copy to Clipboard** to copy the Interfaces table in CSV format.

2. Define the appropriate properties (**“Adding a New Interface” on page 56**).

3. Click **OK**.
**Routes**

The Routes section of the Topology window defines routes between network devices, network addresses, and virtual devices. Some routes are defined automatically based on the interface definitions. You can add new routes as well as delete and change existing routes.

**To add a default route to the routing table:**
1. Click Add Default Route.  
   The Default Gateway window opens.
2. Enter the default route IP address or select the default Virtual Router.
3. Click OK.  
   The default route is added to the routing table.
4. Select the default route and click Edit.  
   The Route Configuration window opens.
5. Configure the settings for the default route and click OK.

**To add a new route to the routing table:**
1. Click Add.  
   The Route Configuration window opens.
2. Configure the Destination IP address and netmask.
3. Configure the next hop IP address or Virtual Router.
4. Optional: Select Propagate route to adjacent virtual devices to “advertise” the route to neighboring virtual devices, and enable connectivity between them.
5. Click OK.

**To change a route:**
1. Select the route.
2. Click Edit.  
   The Route Configuration window opens.
3. Change the settings.
4. Click OK.

**To delete a route:**
1. Select the route.
2. Click Remove.  
   A confirmation window opens.
3. Click OK.

**Topology Calculation**

Select the Calculating topology automatically based on routing information option to let VSX automatically calculate the network topology based on interface and routing definitions. When enabled, VSX creates automatic links, or connectivity cloud objects linked to existing internal or external networks.

- This option is not available in Bridge Mode.
- We recommend that you do not use this option with dynamic routing configurations.
Note - If you wish to enable Anti-Spoofing protection when there are no routes pointing to internal networks, disable the Calculating topology automatically based on routing information option. Modify the appropriate interface definitions to enable Anti-Spoofing.

Deleting a VSX Gateway

When you delete a VSX Gateway object, the system automatically deletes all Virtual Systems and other virtual devices associated with that gateway from the management database.

To delete a VSX Gateway:
1. From the Network Objects tree, right click the VSX Gateway object on the Object Tree and select Delete.
2. Click Yes in the confirmation box.

VSX Gateway Recovery

In the event of a catastrophic VSX Gateway failure, you can use the vsx_util command to restore the VSX Gateway configuration as well as its virtual device configuration.

To restore a VSX Gateway configuration:
1. Reinstall the gateway and configure IP address, net mask and default gateway.
2. Verify that all management interfaces have the same IP addresses as before.
3. From CLI on the management server, run vsx_util reconfigure.

Working with Virtual Systems

This section presents procedures for creating and configuring Virtual Systems. The Virtual System definition process varies somewhat according to the template selected when creating the VSX Gateway.

A typical Virtual System contains two interfaces:

- External interface leading to external networks, a DMZ, or the Internet
- Internal interface leading to internal networks or servers, often by means of a VLAN trunk

VSX supports up to 64 interfaces per virtual device and a total of up to 4096 interfaces per gateway or cluster. The supported interfaces include VLANs and Warp Links.

You can add as many interfaces to a Virtual System as required by your environment, subject to system resource limitations.
The following illustration illustrates an example of a typical VSX Gateway deployment with four Virtual Systems, each containing two interfaces.

Creating a New Virtual System

You use the **Virtual Systems Wizard** to create a new Virtual System. Modify the initial definition and configure advanced options after you complete the wizard.

To start the Virtual System wizard:

1. Open SmartDashboard. If you are using Multi-Domain Security Management, open SmartDashboard from the Domain Management Server in which you are creating the Virtual System.
2. From the **Network Objects** tree, right-click the VSX Gateway and select **VSX > Virtual System**. The **Virtual System Wizard** opens.

**Defining General Properties**

The **General Properties** wizard page defines the Virtual System object and the hosting VSX Gateway.

These are the parameters in this page:

- **Name**: Unique, alphanumeric for the Virtual System. The name cannot contain spaces or special characters except the underscore.
- **VSX Cluster / Gateway**: Select the VSX Gateway that is hosting the Virtual System.
- **Bridge Mode**: Select this option to create a Virtual System in the Bridge Mode.
- **Override Creation Template**: Select this option to override the creation template that was used for the initial configuration of the VSX Gateway.
Defining Network Configuration

The Virtual System Network Configuration page allows you to define internal and external interfaces as well as the IP address topology located behind the internal interface. The process for Virtual System defining network properties varies according to the several factors:

- The **VSX Gateway Creation** template is used to define the VSX Gateway that contains the Virtual System.
- Whether or not you choose to override the default VSX Gateway Creation template. This has the effect of using the **Custom Configuration** template.
- Whether or not you create the Virtual System in the Bridge Mode.

**Note** - Bridge mode is not available for a Virtual System created with the Shared Interface template.

**Shared Interface or Separate Interfaces**

The **Virtual System Network Configuration** page for the Shared Interface and Separate Interfaces templates appears as shown.

To configure the external and internal interfaces:

1. Select the desired interfaces from the appropriate list.
2. If the selected Interface is a VLAN interface, enter the VLAN tag in the appropriate field. This field is not available for non-VLAN interfaces.
3. Enter the IP address and net mask in the appropriate fields. Optionally, enter a default gateway for the external interface.
4. Complete the definition process ("Completing the Definition" on page 46).

**Separate Interfaces in Bridge Mode**

The **Virtual System Network Configuration** page for the Separate Interfaces template in the Bridge Mode appears as shown.

To configure the external and internal interfaces:

1. Select the desired interfaces for the internal and external networks from the appropriate list. If the selected Interface is a VLAN interface, enter the same VLAN tag in both the external and internal **VLAN Tag** fields. This field is not available for non-VLAN interfaces.
2. Define the topology for the internal interface as follows:
   - Select **Not Defined** if you do not wish to define an IP address.
   - Select **Specific** and then select an IP address definition from the list. IP address definitions can be based on object groups or predefined networks that define the topology.
3. If you wish to create a new IP address definition perform the following steps:
   a) Select **Specific** and click **New**.
   b) Select **Group** to define an object group or **Network** to define network properties. The appropriate window appears. Refer to the online help for details regarding either of these options.
4. Enable the **Layer-3 bridge interface monitoring** option if you wish to enable layer 3 network fault detection for this Virtual System.
Configuring VSX

1. Enter an IP address and subnet mask in the designated fields for this Virtual System, which continuously monitors the specified network for faults or connectivity issues. The IP address/subnet should define the network on which the Virtual System resides.

   **Note** - When creating a Virtual System in the bridge mode on an IPSO cluster, you must enable Layer-3 bridge interface monitoring. The IP address to be monitored should reside on a different subnet than the one that handles bridge traffic.

5. Complete the definition process ["Completing the Definition" on page 46].

### Custom Configuration or Override - Non-Bridge Mode

If you used the Custom Configuration template when creating the VSX Gateway, or if you selected the **Override Creation Template** option, it is necessary to manually define the network interfaces and connections. The **Virtual System Network Configuration** page for Custom Configuration appears as shown:

![Virtual System Network Configuration](image)

To configure the external and internal interfaces:

1. In the interface table, define interfaces. You can add new interfaces as well as delete and modify existing interfaces.
   - To add an interface, click **Add**. The **Interface Properties** window opens. Select an interface from the list and define the appropriate properties. Click **Help** for details regarding the various properties and options.

2. Select the **Main IP Address** from the list. This IP address is usually assigned to the external interface and specifies the “real” Virtual System address used when working with NAT or VPN connections.
To make your external IP address routable, select the external interface IP address as the main IP address.

3. Define network routing as appropriate for your deployment. Some routes are automatically defined based on the interface definitions.

For example, you would generally define a default gateway route leading to an external Virtual Router or to the Virtual System external interface.

To add a default route to the Routes table, click Add Default Routes and either enter the default route IP address or select the default Virtual Router. The Route Configuration window opens. Click Help for details regarding the various properties and options.

4. Complete the definition process [“Completing the Definition” on page 46].

Custom or Override in Bridge Mode

If you used the Custom Configuration template when you created the VSX Gateway, or if you selected the Override Creation Template option, and are creating a Virtual System in bridge mode, manually define the network interfaces.

- Define the external and internal interfaces and links to devices in the Interfaces table.
- If the cluster with the bridge is on IPSO, select Enable Layer-3 Bridge Interface Monitoring and define the IP address and net mask. Make sure the IP address to be monitored is on a different subnet than the subnet that handles bridge traffic.

Completing the Definition

Click Next and then Finish to create the Virtual System. Please note that this may take several minutes to complete. A message appears indicating successful or unsuccessful completion of the process.

If the process ends unsuccessfully, click View Report to view the error messages. Refer to the troubleshooting chapter [“VSX Diagnostics and Troubleshooting” on page 157] for further assistance.

Once you create a Virtual System using the Virtual System Wizard, you can modify the topology and all other parameters using the Virtual System Properties window.
Modifying a Virtual System

Once you create a Virtual System using the wizard, you can modify the topology and other properties using the **Virtual System General Properties** window.

**To modify a Virtual System:**

From the **Network Objects** tree, double-click the Virtual System object. The **Virtual System General Properties** window opens.

**Virtual System - General Properties**

The **General Properties** page lets you specify the main IP address and to enable various Check Point products for a Virtual System.

**Virtual System - Topology**

The **Topology** page contains definitions for Virtual System interfaces, routes and Warp Links. Based on these interface settings, VSX automatically creates routes to virtual devices and the VSX Gateway.

Note - If you modify the topology for a specific Virtual System in a cluster environment, the cluster topology is not updated until you install a policy on that Virtual System.

- **Interfaces**: The Interfaces table defines interfaces and links to devices. You can add new interfaces as well as delete and modify existing interfaces.
  
  To add an interface, click **New** and select one of these options:
  
  - **Regular** - Create a new interface
  - **Leads to Virtual Router**
  - **Leads to Virtual Switch**

  The **Interface Properties** window opens. Select the interface from the list and define the appropriate properties. The Modifying an Interface Definition section ("Working with Interface Definitions" on page 56) and the online help provides explanations of the various properties and options.

  Click **Actions > Copy to Clipboard** to copy the **Interfaces** table in CSV format.

- **Routes**: To add a default route to the Routes table, click **Add Default Routes** and either enter an IP address or select a Virtual Router. The **Route Configuration** window opens. Click **Help** for details regarding the various properties and options. You can also add, change and remove routes ("Routes" on page 41).

- **Calculate topology automatically based on routing information**: Enable this option to allow VSX to automatically calculate the network topology based on interface and routing definitions (enabled by default). VSX creates automatic links, or connectivity cloud objects linked to existing internal or external networks.

  - When this option is enabled, you cannot configure the topology using **Topology** tab in the **Interface Properties** window. These options are unavailable on the tab.
  - This option is not available in the Bridge Mode.
  - When employing dynamic routing, it is recommended to disable this option.
• **VPN Domain**: The VPN Domain defines the set of hosts located behind a given Virtual System that communicate via a VPN tunnel with peer Virtual Systems. These options are only available if you selected **VPN** in the *Check Point Products* section on the *General Properties* page.

When including a virtual device as part of a VPN connection, you must specify a VPN Domain. The domain definition specifies Virtual System interfaces that are included in the VPN. You can define a VPN Domain in one of two ways by enabling the appropriate option:

- **All IP Addresses behind gateway based on topology information**: Includes all hosts not located behind an external gateway cluster interface.
- **Manually Defined**: Includes all hosts in the selected network or group.

**Virtual System - NAT > Advanced**

The *NAT > Advanced* page lets you configure NAT rules for packets originating from a Virtual System.

To enable and configure NAT for a Virtual System:

1. Select **Add Automatic Address Translation**.
2. Select a translation method:
   - **Hide**: Hide NAT only allows connections originating from the internal network. Internal hosts can access internal destinations, the Internet and other external networks. External sources cannot initiate a connection to internal network addresses.
   - **Static**: Static NAT translates each private address to a corresponding public address.
3. If you select **Hide**, select one of these options:
   - **Hide behind Gateway** hides the real IP address behind the Virtual System external interface IP address,
     or
   - **Hide behind IP Address** hides the real address behind a virtual IP address, which is a routable, public IP address that does not belongs to any real machine.
4. If you selected **Static NAT**, enter the static IP address in the appropriate field.
5. Select the VSX Gateway from the *Install on Gateway* list.

**Deleting a Virtual System**

To delete a Virtual System, right-click the appropriate Virtual System object on the Object Tree and select **Delete**. Click **Yes** in the confirmation box.

**Working with Virtual Switches**

Virtual Switches provide level-2 connectivity between Virtual Systems and internal or external networks. This section describes how to define and configure a Virtual Switch. As with physical switches, each Virtual Switch maintains a forwarding table containing entries that describe known networks and directions for reaching them.
You can define Virtual Switches for external and internal communications.

This figure shows a typical deployment using a Virtual Switch for external connections and a VLAN trunk leading to the internal, protected network.

**Modifying Virtual Switches**

Once you create a Virtual Switch using the wizard, you can modify the topology and other properties using the *Virtual Switch General Properties* window.

To modify a Virtual Switch:

From the *Network Objects* tree, double-click the Virtual Switch object. The *Virtual Switch General Properties* window opens.

**Virtual Switch - General Properties**

The *General Properties* page allows you to add comments and change the icon color as displayed in SmartDashboard.

**Virtual Switch - Topology**

The *Topology* page defines Virtual Switch interfaces. You can only modify the single defined interface. Warp interfaces cannot be modified from this window.

To add an interface, click *New*. The *Interface Properties* window opens. Select an interface from the list and define the IP address, net mask and other properties as required.

Click *Actions > Copy to Clipboard* to copy the *Interfaces* table in CSV format.

Refer to the Modifying an Interface Definition section (“Working with Interface Definitions” on page 56) or the online help for details regarding the various properties and options.
Deleting a Virtual Switch

Remove all Virtual System connections before you delete a Virtual Switch.

To delete a Virtual Switch, right-click the appropriate Virtual Switch object in the Object Tree and select Delete. Click Yes in the confirmation box.

Creating a New Virtual Switch

Use the Virtual Switch Wizard to create a new Virtual Switch. You can modify the initial definition and configure advanced options after completing the wizard.

To create a new Virtual Switch:
1. Open SmartDashboard.
   If you are using Multi-Domain Security Management, open SmartDashboard from the Domain Management Server in which you are creating the Virtual Switch.
2. From the Network Objects tree, right-click Check Point and select VSX > Virtual Switch.
   The General Properties page of the Virtual Switch Wizard opens.
3. Enter the name of the Virtual Switch.
4. Select the VSX Gateway or cluster to which the Virtual Switch connects.
5. Click Next.
6. Click Add.
   The Add Interface window opens.
7. Configure the interface on the Virtual Switch.
8. Click OK and then click Next.
9. Click Finish.

Working with Virtual Routers

This section describes how to define and configure a Virtual Router. As with physical routers, each Virtual Router maintains a routing table containing entries that describe known networks and directions on how to reach them.
You can define Virtual Routers for both external and internal communications. A Virtual Router that connects to external networks, including a DMZ and the Internet, are referred to as an external Virtual Router. A Virtual Router that connects to internal, protected networks is known as an internal Virtual Router.

An external Virtual Router functions as the external gateway for Virtual Systems, allowing them to share a single secure physical interface leading to external networks and the Internet.

In this scenario, VSX creates Warp interfaces between the Virtual Systems and both Virtual Routers. Note that the external Virtual System interfaces are defined as unnumbered interfaces.

An internal Virtual Router typically connects with one interface leading to internal networks through a switch with additional Warp Links leading to other Virtual Systems located in the VSX Gateway.

After you create a new Virtual Router, add new interfaces to the Virtual Systems to connect to the Virtual Router.
Creating a New Virtual Router

Use the Virtual Router Wizard to create a new Virtual Router. You can modify the initial definition and configure advanced options after you complete the wizard.

On interfaces and routes, you can select the Propagate route to adjacent virtual devices option to broadcast the IP address to neighboring virtual devices. This option enables connectivity with these virtual devices.

To create a Virtual Router:
1. Open SmartDashboard.
   If you are using Multi-Domain Security Management, open SmartDashboard from the Domain Management Server in which you are creating the Virtual System.
2. From the Network Objects tree, right-click the VSX Gateway and select VSX > Virtual Router. The General Properties page of the Virtual Router Wizard opens.
3. Enter the name of the Virtual Router.
4. Select the VSX Gateway or cluster to which the Virtual Router connects.
5. Click Next.
6. From the Interfaces section, click Add. The Add Interface window opens.
7. Configure the interface on the Virtual Router.
8. Click OK.
9. From the Routes section, click Add. The Route Configuration window opens.
10. Configure the network routes.
11. Click OK.
12. Optional: Click Add Default Route and configure the default route.
13. Click Next and then click Finish.

Modifying a Virtual Router Definition

Once you create a Virtual Router using the wizard, you can modify the topology and other properties using the Check Point Virtual Router window. This window also allows you to configure many advanced features and options that are not available in the wizard.

To work with a Virtual Router definition, double-click the Virtual Router object in the Object tree. The Check Point Virtual Router window opens, displaying the General Properties page.

Virtual Router - General Properties

The General Properties page enables you change the Virtual Router IP address as well as to add comments and change the icon color as displayed in SmartDashboard.

Virtual Router - Topology

The Virtual Router Network Configuration page defines the network topology for the Virtual Router. For an external interface, you define one or more shared external interfaces and a default gateway.
Topology is defined by these properties:

- **Interfaces**: Add new interfaces, or modify or delete existing interfaces.
  To add an interface, click **New**. The **Interface Properties** window opens. Select an interface from the list and define the IP address, net mask and other properties ("Working with Interface Definitions" on page 56).

- **Routes**: Add network routes between this Virtual Router, Virtual Systems, external network devices and network addresses. Some Warp Link routes are defined automatically and cannot be modified or deleted. You can manually add new routes as well as delete and modify non-Warp Link routes.

- **Add Default Route**: Define the default route as an IP address or Virtual System.

- **Advanced Routing**: Configure source-based routing ("Working with Source-Based Routing" on page 53) rules.

### Deleting a Virtual Router

You cannot delete a Virtual Router if it is still connected to a Virtual System. Remove all Virtual Router connections before deleting.

To delete a Virtual Router, right-click the appropriate Virtual Router object on the Object Tree and select **Delete**. Click **Yes** in the confirmation box.

### Working with Source-Based Routing

Source-based routing directs traffic to a specific destination based on the source IP address or a combination of the source and destination IP addresses. Rules defining Source-based routing take precedence over ordinary destination-based routing rules.

This section describes how to configure sourced-based routing rules when working in a VSX environment. The procedures for defining source-based rules are the same for Virtual Routers in both VSX Gateways and VSX clusters.
Defining Source-Based Routing Rules


To define source-based routing rules:

1. Open the appropriate internal Virtual Router definition and select the Topology page.
2. Click Advanced Routing.
   The Advanced Routing Rules window opens.
   Note: The highlighted rule is based on a source and a destination address, as compared to the preceding rules, which are based on a source address only.
3. Click Add, to define a new rule or Edit, to change an existing rule.
   The Add/Edit Route Rule window opens.
   Define the properties:
   • Source IP Address and Net Mask
   • Destination IP Address and Net Mask
   • Next Hop Gateway: Select a Virtual System from the list.

Defining Source-Based Routing Rules

Use the Advanced Routing Rules window to define source-based routing rules.

To define source-based routing rules:

1. Open SmartDashboard.
2. From the Network Objects tree, right-click the Virtual Router and select Edit.
   The General Properties window opens.
   From the navigation tree, select Topology.
3. Click Advanced Routing.
   The Advanced Routing Rules window opens.
4. Click Add, to define a new rule or Edit, to change an existing rule.
   The Add/Edit Route Rule window opens.
5. Define these settings:
   • Source IP Address and Net Mask
   • Destination IP Address and Net Mask
   • Next Hop Gateway
6. Click OK.

CoreXL for Virtual Systems

CoreXL creates multiple firewall instances that are, in reality, independent firewalls. You can use CoreXL to increase the performance of the VSX Gateway on an open server or appliance with multiple cores. You can also assign each instance to a group of CPU cores with the fw ctl affinity command ("CoreXL Commands (fw ctl affinity)" on page 164).

You configure firewall instances differently for the VSX Gateway (VS0) than for other Virtual Systems.
• VSX Gateway - Use the CLI to configure the number of instances.
- **Other Virtual Systems** - Use SmartDashboard to configure the number of instances.

You can configure multiple instances for each Virtual System. When you change the number of firewall instances on a Virtual System, there is some downtime for that Virtual System.

⚠ Important - Each firewall instance that you create uses additional system memory. A Virtual System with five instances would use approximately the same amount of memory as five separate Virtual Systems.

The number of IPv6 instances cannot exceed the number of IPv4 instances. For more about IPv6 instances and VSX, go to sk97997 http://supportcontent.checkpoint.com/solutions?id=sk97997.

For more about configuring CoreXL, see the *R76 Performance Tuning Administration Guide* http://supportcontent.checkpoint.com/documentation_download?ID=22917.

### Configuring CoreXL on a VSX Gateway

Use the `cpconfig` command to configure CoreXL on the VSX Gateway (VS0). The number of instances for the VSX Gateway is limited to the physical number of cores on the server or appliance.

**To configure the number of instances on the VSX Gateway:**

1. From the CLI, run `cpconfig`.
2. Select **Configure Check Point CoreXL**.
3. Make sure that CoreXL is enabled.
4. Configure the number of firewall instances for the VSX Gateway.
5. Exit `cpconfig`.

Note - It is not necessary to reboot the VSX Gateway after you configure CoreXL.

### Configuring CoreXL on Virtual Systems

Use SmartDashboard to configure the number of firewall instances on the Virtual Systems. You can assign up to 8 instances on a Virtual System. The number of instances is not limited by the physical cores on the VSX Gateway server or appliance.

You can assign the number of IPv6 instances. It must be less or equal to the number of IPv4 instances. The number of IPv6 instances may be zero. IPv6 virtual instances will only be enabled if an IPv6 address is configured for that Virtual System.

We recommend that you use the number of instances for each Virtual System according to the expected network traffic on the Virtual System. Configuring unnecessary instances will have an impact on performance.

We recommend that you do not configure more instances than the total number of physical cores on the VSX Gateway server or appliance.

**To configure CoreXL on a Virtual System:**

1. Open SmartDashboard.
2. From the **Network Objects** tree, double-click the **Virtual System**. The **Virtual System General Properties** window opens.
3. From the navigation tree, select **CoreXL**.
4. Select the number of firewall instances for the Virtual System.
5. Click OK.

Dynamic Routing for virtual devices

This section presents procedures for configuring dynamic routing for Virtual Systems and Virtual Routers. The virtual devices can use dynamic routing protocols to communicate and distribute routes amongst themselves and with external routers and other devices. VSX uses the Gaia routing daemon (routed).

You can configure dynamic routing for each of these virtual devices:
- Virtual System
- Virtual Router

Each of these virtual devices has its own dynamic routing instance and configuration file. Use the same procedures to configure the dynamic routing protocols for Warp Links as regular interfaces. You can also configure dynamic routing separately on each cluster member.

To configure dynamic routing for a virtual device:
1. Set the context to the virtual device. From the CLI, run `set virtual-system <vsid>`
2. Run the commands to configure the dynamic routing daemon for the virtual device.

   Note - When you run a dynamic routing command, use the TAB key to show the interfaces that you can enable.

Working with Interface Definitions

All VSX Gateways and Virtual Routers and Virtual Switches contain at least one interface definition. Typically, you define the interfaces during the process of configuring the topology for a given object. Warp interfaces, however, are created automatically based on virtual device definitions and their topology. You cannot modify or delete a Warp interface.

Adding a New Interface

All VSX Gateways and Virtual Routers and Virtual Switches have at least one interface definition. Typically, you define the interfaces when you configure the topology for a given object. Warp interfaces are created automatically based on virtual device definitions and their topology. You cannot change or delete a Warp interface.

The procedure and options for defining an interface vary according to the object and the network topology. Some properties and pages are not available for certain interface definitions.

To add a new interface:
1. Open the Gateway Properties window for the virtual device.
2. From the navigation tree, click Topology.
   
   The Topology page opens.
3. From the Interfaces section, click New and select one of these options:
   - Regular
   - Leads to Virtual Router
• **Leads to Virtual Switch**
  The Interface Properties window for the selected option opens.

**Configuring Connection Properties - General**

The General tab defines the network connections associated with an interface.

One or more of these properties show, depending on the context.

- **Interface**: Select a physical interface from the list (physical interfaces only).
- **VLAN Tag**: VLAN tag associated with the defined interface.
- **IP Address** and **Net Mask**: IP address and net mask of the device associated with the interface.
- **Propagate route to adjacent virtual devices**: Enable to “advertise” the associated device to neighboring devices, thereby enabling connectivity between them. The Route Propagation section (See "Route Propagation" on page 28) provides additional details.
- **MTU**: Maximum transmission unit size in bytes (default = 1,500).

**Configuring Connections Leading to Virtual Routers and Switches**

The General tab for interface connections leading to Virtual Routers or Virtual Switches contains connection properties specific to those virtual devices.

- **Leads to**: Select a Virtual Router or Virtual Switch.
- Enter the dedicated Virtual System **IP address** for this interface.
- The **Net Mask** property is always defined as 255.255.255.255 and cannot be modified.
- **Propagate route to adjacent virtual devices**: Enable to “advertise” the associated device to neighboring devices, thereby enabling connectivity between them. The Route Propagation section (See "Route Propagation" on page 28) provides additional details.
- **MTU**: Maximum transmission unit size in bytes (default = 1,500).

**Configuring Interface Topology**

For some interface types, you can change some or all of these topology properties:

- **External**: The interface leads to external networks or to the Internet.
- **Internal**: The interface leads to internal networks or a DMZ, and includes these properties:
  - **Not Defined**: IP routing is not defined for this device.
  - **Network**: Routing is defined by the IP and net mask defined in General Properties.
  - **Specific**: Routing is defined by a specific network or network group.
  - **Interface leads to DMZ**: Defines an interface as leading to a DMZ, which isolates a vulnerable, externally accessible resource from the rest of a protected, internal network.

**Configuring Anti-Spoofing**

Attckers can gain access to protected networks by falsifying or “spoofing” a trusted source IP address with high access privileges. It is important to configure Anti-Spoofing protection for VSX Gateways and Virtual Systems, including internal interfaces. You can configure Anti-Spoofing for an interface, provided that the topology for the interface is properly defined.

If you are using dynamic routing, disable the Calculate topology automatically based on routing information option, and manually configure the topology of the Virtual System.
To enable Anti-Spoofing for an interface:

1. From the Topology tab in the Interface Properties window, select Perform Anti-Spoofing based on interface topology.
2. Configure the tracking options.

**Configuring Multicast Restrictions**

IP multicasting applications send one copy of each datagram (IP packet) and address it to a group of computers that wish to receive it. Multicast restrictions allow you to define rules that block outbound datagrams from specific multicast groups (IP address ranges). You can define multicast access restrictions for physical and Warp interfaces in a VSX environment.

According to RFC 1112, you can only use an IP address in between 224.0.0.0 and 239.255.255.255.

To enable multicast restrictions:

1. From the Multicast Restrictions tab in the Interface Properties window, select Drop multicast packets by the following conditions.
2. Select one of the following restriction types:
   - Drop multicast packets whose destination is in the list
   - Drop all multicast packets except those whose destination is in the list
3. Click Add.
   The Add Object window opens.
4. Click New > Multicast Address Range.
   The Multicast Address Range Properties window opens.
5. Configure these settings:
   - Name
   - Type
   - If you selected IP Address Range, enter the First and Last IP addresses.
6. Click OK.
7. From the Interface Properties window, select a tracking option.
8. Click OK and close the General Properties window.
9. Add a rule to the Rule Base that allows traffic for the specified multicast groups and install the policy.

**Changing an Interface Definition**

This section presents procedures for modifying existing interface definitions and related features.

**Changing an Interface**

Interfaces definitions are always associated with a Virtual Gateway or a Virtual System definition.

To work with an existing interface definition:

1. Double-click the interface in the Interfaces section.
2. In the Interface Properties window, define the interface properties ("Adding a New Interface" on page 56).
Deleting an Interface

To delete an interface:

1. From the Topology page, select the interface and click Delete.
2. Click OK.

Working with Authentication

Supported Authentication Schemes

Authentication schemes employ user names and passwords to identify valid users. Some schemes are maintained locally, storing user names and passwords on the VSX Gateway, while others store authentication information on an external authentication server. Some schemes, such as SecurID, are based on providing a one-time password.

All of the schemes can be used with users defined on an LDAP server. For additional information on configuring a Security Gateway to integrate with an LDAP server, refer to the User Directory (LDAP) and User Management section in the R76 Security Management Administration Guide.

Check Point Password

VSX stores a static password for each user in the management server database. No more software is required.

Operating System Password

VSX can authenticate users by means of a user name and password defined on the management server operating system. You can also use passwords stored in a Windows domain. No additional software is required.

RADIUS

Remote Authentication Dial In User Service (RADIUS) is an external, server-based authentication protocol that provides authentication services using the UDP protocol.

TACACS, TACACS+

Terminal Access Controller Access Control System (TACACS) is an external, server-based authentication protocol that provides verification services using the TCP protocol. TACACS+ is an enhanced version of the TACACS that supports additional types or authentication requests and response codes.

SecurID

SecurID requires users to possess a token authenticator and to supply a password. Token authenticators generate one-time passwords that are synchronized to an RSA ACE/Server. Hardware tokens are key-ring or credit card-sized devices, while software tokens reside on the PC or device from which the user wants to authenticate. All tokens generate a random, one-time use access code that changes approximately every minute. When a user attempts to authenticate to a protected resource, the one-time use code must be validated by the ACE/Server.
Configuring SecurID ACE/Server

These are the options to enable connectivity between Virtual Systems and a SecurID ACE/Server:

- **Shared configuration**: All authentication servers are accessible by all Virtual Systems through the VSX Gateway. The Virtual Systems on each VSX Gateway use the same encryption key. This is the default option.

- **Private configuration**: Authentication servers are accessed directly by the Virtual System and use the Virtual System cluster IP address as the source address. Each Virtual System uses a separate encryption key. For High Availability configurations, the Virtual Systems on different VSX cluster members use the same encryption key.

  **Note** - You can configure authentication for more than one ACE/Server in private mode. Contact Check Point Technical Support for more information [http://supportcenter.checkpoint.com](http://supportcenter.checkpoint.com).

The SecurID ACE/Server sends a shared key (called a "node secret") to its peer ACE/Clients. This key is unique per IP address, and is sent when it connects to the ACE/Server for the first time.

**Configuring Shared Authentication**

Configure shared authentication so that all the Virtual Systems on the VSX Gateway use the same encryption key to authenticate to the remote SecurID/ACE server. Each cluster member uses a different encryption key and node secret file.

The SecurID encryption key is stored in the `sdconf.rec` file. When you generate the `sdconf.rec` file, use the MIP (Member IP address) of a VSX Gateway interface that connects to the ACE/Server.

The first time that a Virtual System connects and attempts to authenticate to the ACE/Server, the server sends the node secret file (`securid`) to that Virtual System. Copy the node to all the other Virtual Systems.

**To generate an `sdconf.rec` file:**

1. From the ACE/Server, generate the `sdconf.rec` file with the VSX Gateway MIP.
2. Do the previous step again for each cluster member using the VSX Gateway MIP.
   - For example, a cluster with three VSX Gateways and each member has five Virtual Systems. Generate three `sdconf.rec` files, one for each cluster member.

**To configure shared authentication:**

   a) Open SmartDashboard.
   b) From the Network Objects tree, double-click the Virtual System. The Virtual Systems General Properties window opens.
   c) From the navigation tree, select Other > Legacy Authentication.
   d) Make sure that SecurID and Shared are selected.
   e) Click OK.
   
   Do all of the previous steps for each Virtual System.
f) Install the policy on the Virtual Systems.

2. From the VSX Gateway CLI, for each Virtual System create the sdopts.rec file that contains the MIP.
   a) Enter Expert mode and change the context to the Virtual System. Run
      ```
      # vsenv <vsid>
      ```
   b) Create the file, $VAR_ACE/sdopts.rec
      For VS0, create the file /var/ace/sdopts.rec
   c) From a text editor, add this parameter to the sdopts.rec file.
      ```
      CLIENT_IP=<MIP>
      ```
      
      <MIP> is the Member IP address of the VSX Gateway.

3. For each Virtual System, copy the same encryption key file, sdconf.rec, to $VAR_ACE.
   For VS0, copy the file to /var/ace.
   On 61000/41000 Security Systems, copy the same encryption key file sdconf.rec to $FWDIR/conf in the context of that Virtual System.

4. For cluster configurations, do all of the previous steps for each cluster member.

5. For cluster configurations, on the Security Management Server of the VSX Gateway make sure that Hide NAT is disabled.
   a) Open the table.def file.
      - Gaia, SecurePlatform, IPSO - $FWDIR/lib/table.def
      - Windows - %FWDIR%/lib\table.def
   b) Make sure that the no_hide_services_ports parameter contains UDP 5500.
      Sample parameter with Hide NAT disabled:
      ```
      no_hide_services_ports = { <500, 17>, <259, 17>, <1701, 17>, <123, 17>, <5500, 17> };
      ```
   c) Save the file.
   d) From SmartDashboard, install the policy on the VSX Gateway.

To distribute the node secret to the Virtual Systems:

1. Authenticate to the gateway with a SecurID ACE/Server user account.
   The ACE/Server sends the node secret file to the gateway.

2. Search each Virtual System to locate the node secret file, securid.
   - For VS0, search in /var/ace.
   - For other Virtual Systems, search in $VAR_ACE.
   - On 61000/41000 Security Systems, search in $FWDIR/conf in the context of each Virtual System.

3. Copy the securid file to $VAR_ACE.
   For VS0, copy the file to /var/ace.
   On 61000/41000 Security Systems, copy the securid file to $FWDIR/conf in the context of each Virtual System.
For cluster configurations, for each cluster member:

- Locate a Virtual System that is active on that member and do all the previous steps.
- If there are no active Virtual Systems on that member, fail-over to the cluster member and then do all the previous steps.

**Configuring Private Authentication**

Configure private authentication so that the active and standby Virtual Systems use the same encryption key and node secret file to authenticate to the remote SecurID ACE/Server.

The SecurID encryption key is stored in the `sdconf.rec` file. When you generate the `sdconf.rec` file, use the VIP (Virtual IP address) of the Virtual System interface that connects to the ACE/Server.

The first time that a VSX Gateway connects to the ACE/Server, the server sends the node secret file (securid) to that VSX Gateway. Copy the node to all the other VSX Gateways.

**To generate an `sdconf.rec` file:**

1. From the ACE/Server, generate the `sdconf.rec` file with the Virtual System VIP.
2. Do the previous step again for each Virtual System on the VSX Gateway.
   - For example, a cluster with three VSX Gateways and each member has five Virtual Systems.
   - Generate five `sdconf.rec` files, one for each Virtual System.

**To configure private authentication:**

   a) Open SmartDashboard.
   b) From the Network Objects tree, double-click the VSX Gateway.
      - The **VSX Gateway General Properties** window opens.
   c) From the navigation tree, select **Other > Authentication**.
   d) Make sure that **SecurID** and **Private** are selected.
   e) Click **OK**.
      - Do all of the previous steps for each Virtual System.
   f) Install the policy on the Virtual Systems.
2. From the VSX Gateway CLI, for each Virtual System create the `sdopts.rec` file that contains the VIP of that Virtual System.
   a) Enter Expert mode and change the context to the Virtual System. Run: `# vsenv <vsid>`
   b) Create the file, `$VAR_ACE/sdopts.rec`
      - For VS0, create the file `/var/ace/sdopts.rec`
      - On 61000/41000 Security Systems, create the file `$FWDIR/conf/sdopts.rec` in the context of each Virtual System.
   c) From a text editor, add this parameter to the `sdopts.rec` file.
      ```
      CLIENT_IP=<VIP>
      <VIP> is the Virtual IP address of the Virtual System.
      ```
3. For each Virtual System, copy the encryption key file, sdconf.rec, to $VAR_ACE.
   Each Virtual System on the VSX Gateway uses a different sdopts.rec file.
   For VSO, copy the file to /var/ace.
   On 61000/41000 Security Systems, copy the file to $FWDIR/conf/ in the context of each Virtual System.

4. For cluster configurations, do all of the previous steps for each cluster member.

5. For cluster configurations, on the Security Management Server make sure that Hide NAT is enabled.
   For Multi-Domain Server, use the Domain Management Server that manages the Virtual System.
   a) Open the table.def file.
      - Gaia, SecurePlatform, IPSO - $FWDIR/lib/table.def
      - Windows - %FWDIR%/lib/table.def
   b) Make sure that the no_hide_services_ports parameter DOES NOT contain UDP 5500.
      Sample parameter with Hide NAT enabled:
      no_hide_services_ports = { <500, 17>, <259, 17>, <1701, 17>, <123, 17> };
   c) Save the file.
   d) From SmartDashboard, install the policy on the Virtual Systems.

To distribute the node secret to Virtual Systems in a VSX cluster:
1. Authenticate to the Virtual System on the gateway with a SecurID ACE/Server user account.
   The ACE/Server sends the node secret file to the gateway.
2. Locate the cluster member of the active Virtual System.
3. From that cluster member, copy the securid file to the same Virtual System on the other members.
   - For VSO, copy the file to /var/ace.
   - For other Virtual Systems, copy the file to $VAR_ACE.
   - On 61000/41000 Security Systems, copy the file to $FWDIR/conf/ in the context of each Virtual System.
4. Do all of the previous steps for each Virtual System.

Configuring RADIUS or TACACS/TACACS+

These are the options to enable connectivity between Virtual Systems and a RADIUS or TACACS/TACACS+ server:
- **Shared configuration**: All authentication servers are accessible by all Virtual Systems through the VSX Gateway. This is the default option.
- **Private configuration**: Authentication servers are accessed directly by the Virtual System and use the Virtual System cluster IP address as the source address.

For Multi-Domain Server configurations, make sure that you configure the SecurID or Remote Authentication settings of the Domain Management Server that manages the Virtual Systems.
Configuring Shared Authentication

Configure shared authentication so that all the Virtual Systems on the VSX Gateway authenticate to the remote RADIUS or TACACS/TACACS+ server.

To configure shared authentication for RADIUS or TACACS/TACACS+:

   a) Open SmartDashboard.
   b) From the Network Objects tree, double-click the Virtual System.
      The Virtual Systems General Properties window opens.
   c) From the navigation tree, select Other > Authentication.
   d) Make sure that RADIUS or TACACS and Shared are selected.
   e) Click OK.
      Do all of the previous steps for each Virtual System.
   f) Install the policy on the Virtual Systems.

2. For cluster configurations, on the Security Management Server of the VSX Gateway make sure that Hide NAT is disabled.
   a) Open the table.def file.
      - Gaia, SecurePlatform, IPSO - $FWDIR/lib/table.def
      - Windows - %FWDIR%/lib/table.def
   b) Make sure that the no_hide_services_ports parameter contains the UDP ports for RADIUS or TACACS, or the TCP ports for TACACS+. The default ports are:
      - RADIUS - 1645
      - TACACS/TACACS+ - 49
      Sample RADIUS parameter with Hide NAT disabled:
      no_hide_services_ports = { <49, 6>, <49, 17>, <500, 17>, <259, 17>, <1701, 17>, <123, 17>, <1645, 17> };
   c) Save the file.
   d) From SmartDashboard, install the policy on the VSX Gateway.

Configuring Private Authentication

For private configurations, the active and standby Virtual Systems use the same encryption key to authenticate to the remote RADIUS or TACACS/TACACS+ server.

For High Availability configurations, make sure that the active and standby Virtual Systems on each cluster member use the same VIP.

To configure private authentication:

   a) Open SmartDashboard.
   b) From the Network Objects tree, double-click the VSX Gateway.
      The VSX Gateway General Properties window opens.
c) From the navigation tree, select Other > Authentication.
d) Make sure that RADIUS or TACACS and Private are selected.
e) Click OK.

Do all of the previous steps for each Virtual System.
f) Install the policy on the Virtual Systems.

2. For cluster configurations, on the Security Management Server make sure that Hide NAT is enabled.
For Multi-Domain Server, use the Domain Management Server that manages the Virtual System.
a) Open the table.def file.
   - Gaia, SecurePlatform, IPSO - $FWDIR/lib/table.def
   - Windows - %FWDIR%/lib/table.def
b) Make sure that the no_hide_services_ports parameter DOES NOT contain the UDP ports for RADIUS or TACACS, or the TCP ports for TACACS+.
   The default ports are:
   - RADIUS - 1645
   - TACACS/TACACS+ - 49
   Sample parameter with Hide NAT enabled:
   
   no_hide_services_ports = { <500, 17>, <259, 17>, <1701, 17>, <123, 17> };

c) Save the file.
d) From SmartDashboard, install the policy on the Virtual Systems.

Tracking Activity with SmartView Monitor

SmartView Monitor is the Graphical User Interface application that displays the status of all Security Gateways, VSX Gateways and virtual devices. SmartView Monitor displays a snapshot of installed Check Point products including VSX. For each VSX Gateway, VSX cluster, or virtual device, SmartView Monitor shows a full range of information including, operating system, CPU, and memory. You can also enable the SmartView Monitor Software Blade on a Virtual System to show detailed information about the network traffic.

For more information on using SmartView Monitor, refer to the R76 SmartView Monitor Administration Guide http://supportcontent.checkpoint.com/documentation_download?ID=22925.

SmartView Monitor connects to and validates each Virtual System as an independent gateway. If one Virtual System is down, this information will be reflected in SmartView Monitor even though the other Virtual Systems on the VSX Gateway or VSX cluster are functioning normally.

Client/Session Authentication

VSX supports the following client/session authentication schemes:

- Client authentication over TELNET (on port 259)
• Client authentication over HTTP/HTTPS (on port 900)

For a complete description of these features, see the R76 IPS Administration Guide

Configuring Client/Session Authentication

In a VSX environment, you configure Client/Session authentication settings by manually editing the $FWDIR/conf/cpauthd.conf file, located on the VSX Gateway.

Note - The VSX client/session authentication procedure is different than the one for Security Gateways.

You must configure client/session for the VSX Gateway. These settings apply, by default, to all Virtual Systems located on the gateway.

You can optionally configure client/session authentication for specific Virtual Systems. Virtual System specific settings override the default settings for that Virtual System only. Virtual Systems that do not have their own settings inherit the default settings.

Configuring Authentication for the VSX Gateway

To configure client/session authentication for the VSX Gateway:

2. From the VSX Gateway, use a text editor to open $FWDIR/conf/cpauthd.conf.
3. Add or configure the appropriate parameters.
4. Run these commands:
   cpwd_admin stop -name FWD -path "$FWDIR/bin/fw" -command "fw kill fwd"
   cpwd_admin start -name FWD -path "$FWDIR/bin/fwd" -command "fwd"

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>clauth_port</td>
<td>259</td>
<td>The TCP port on which client authentication over TELNET is done.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = Client authentication over TELNET is disabled.</td>
</tr>
<tr>
<td>clauth_http_port</td>
<td>900</td>
<td>The TCP port on which client authentication over HTTP/HTTPS is done.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = Client authentication over HTTP/HTTPS is disabled.</td>
</tr>
<tr>
<td>clauth_http_ssl</td>
<td>0</td>
<td>0 = HTTPS client authentication is disabled.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = HTTPS client authentication is enabled.</td>
</tr>
<tr>
<td>clauth_http_nickname</td>
<td>none</td>
<td>Specifies the certificate nickname when client authentication is performed over HTTPS.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This attribute must match the Virtual System certificate nickname as configured using SmartDashboard (Virtual System &gt;VPN &gt;Certificate List).</td>
</tr>
</tbody>
</table>
Configuring Authentication for Specific Virtual Systems

To configure client/session authentication for the VSX Gateway:

1. Backup $FWDIR/CTX/CTX#/conf/cpauthd.conf, where CTX# refers to the specific Virtual System directory.
2. Delete the original $FWDIR/CTX/CTX#/conf/cpauthd.conf
3. Copy $FWDIR/conf/cpauthd.conf to $FWDIR/CTX/CTX#/conf/cpauthd.conf
4. Open a text editor, and add or configure the appropriate parameters.
5. Run these commands:
   
   ```
   cpwd_admin stop -name FWD -path "$FWDIR/bin/fw" -command "fw kill fwd"
   cpwd_admin start -name FWD -path "$FWDIR/bin/fwd" -command "fwd"
   ```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
</table>
| clauth_port         | 259           | The TCP port on which client authentication over TELNET is performed.  
|                     |               | 0 = Client authentication over TELNET is disabled. |
| clauth_http_port    | 900           | The TCP port on which client authentication over HTTP/HTTPS is performed.  
|                     |               | 0 = Client authentication over HTTP/HTTPS is disabled. |
| clauth_http_ssl     | 0             | 0 = HTTPS client authentication is disabled.  
|                     |               | 1 = HTTPS client authentication is enabled. |
| clauth_http.nickname| none          | Specifies the certificate nickname when client authentication is performed over HTTPS.  
|                     |               | This attribute must match the Virtual System certificate nickname as configured using SmartDashboard [Virtual System >VPN >Certificate List]. |

**Notes**

- cpauthd.conf is used instead of $FWDIR/conf/fwauthd.conf on a non-VSX Gateway.
- All Virtual Systems (other than the default Virtual System) are assigned a symbolic link in the $FWDIR/CTX/CTX#/conf/cpauthd.conf file. This link points to $FWDIR/conf/cpauthd.conf, where CTX# refers to the specific Virtual System directory.

Working with Network Address Translation

This section describes the process for using Network Address Translation (NAT) in a VSX deployment. The procedures described in this section assume that the reader is familiar with NAT concepts and their implementation in Check Point products. For more about NAT, see the Network Address Translation chapter in the R76 Firewall Administration Guide.

VSX supports NAT for Virtual Systems much in the same manner as a physical firewall. When a NAT enabled [Static or Hide] Virtual System connects to a Virtual Router, the translated routes are automatically forwarded to the appropriate Virtual Router.
Configuring NAT

You configure NAT using the **NAT** page in the **Virtual System** window. Hide or Static NAT addresses configured in this manner are automatically forwarded to the Virtual Router to which the Virtual System is connected. Alternatively, you can manually add NAT routes on the **Topology** page in the **Virtual Router** window.

To configure NAT for a Virtual System:

1. Open the **Gateway Properties** window for the virtual device.
2. From the navigation tree, click **NAT > Advanced**.
   The **Advanced** page opens.
3. Select **Add Automatic Address Translation**.
4. Select a **Translation method**.
   - **Hide**: Hide NAT only allows connections originating from the internal network. Internal hosts can access internal destinations, the Internet and other external networks. External sources cannot initiate a connection to internal network addresses.
   - **Static**: Static NAT translates each private address to a corresponding public address.
5. If you select **Hide**, select one of the following options:
   - **Hide behind Gateway** hides the real address behind the VSX Gateway external interface address. This is equivalent to hiding behind the address 0.0.0.0.
   - **Hide behind IP Address** hides the real address behind a virtual IP address, which is a routable, public IP address that does not belong to any real machine.
6. If you selected **Static**, enter the static IP address.
7. From the **Install on Gateway** list, select the VSX Gateway.
8. Click **OK**.

Using Application Control and URL Filtering with VSX

When you configure Virtual Systems to use the Application Control and URL Filtering Software Blades, you must configure the VSX Gateway to connect to the Application Control and URL Filtering Database. Make sure that the VSX Gateway (VS0) can connect to the Internet, because communication for updates and URL categories is only done from this Virtual System.

To enable Application Control and URL Filtering Categories on Virtual Systems:

1. From the **Network Object** tree, double-click the VSX Gateway.
2. From the navigation tree, select **Topology > Proxy**.
3. Configure the proxy settings, and click **OK**.
4. Enable Application Control and URL Filtering on the Virtual System:
   a) From the **Network Object** tree, double-click the Virtual System.
   b) In the **Network Security** section, select **Application Control and URL Filtering**, and then click **OK**.
5. Do the previous step again for all the Virtual Systems that are using Application Control and URL Filtering.
6. Select the **Application Control and URL Filtering** tab and configure the policies.
7. Install the policies on the VSX Gateway and Virtual Systems.
Using Anti-Bot and Anti-Virus with VSX

When you configure Virtual Systems to use the Anti-Bot and Anti-Virus Software Blades, you must configure the VSX Gateway to connect to the Anti-Bot and Anti-Virus Database. Make sure that the VSX Gateway (VS0) can connect to the Internet.

To enable Application Control and URL Filtering Categories on Virtual Systems:

1. From the Network Object tree, double-click the VSX Gateway.
2. From the navigation tree, select Topology > Proxy.
3. Configure the proxy settings, and click OK.
4. Enable Anti-Bot and Anti-Virus on the Virtual System:
   a) From the Network Object tree, double-click the Virtual System.
   b) In the Network Security section, select Anti-Bot and Anti-Virus, and then click OK.
5. Do the previous step again for all the Virtual Systems that are using Anti-Bot and Anti-Virus.
6. Select the Anti-Bot and Anti-Virus tab and configure the policies.
7. Install the policies on the VSX Gateway and Virtual Systems.

Licensing VSX

Check Point software is activated with a license key. To obtain a license key, register the certificate key (that appears on the back of the software media pack) with the Check Point User Center. The certificate key is used to generate a license key for the products that you are either evaluating or purchasing. To purchase the required Check Point products, contact your reseller. Check Point software that has not yet been purchased functions for 15 days only.

VSX Gateway/Cluster Member Licenses

Each VSX Gateway or VSX cluster member requires its own license, bound to the gateway or cluster member IP address. Each gateway / cluster license covers a predefined number of Virtual Systems (3, 10, 25, and 50) and these licenses are cumulative. The VSX licenses are applied in addition to the Security Gateway license [container and Software Blades].

Upgrading Licenses

Before upgrading a gateway or Security Management Server to R76, you need to have a valid support contract that includes software upgrade and major releases registered to your Check Point User Center account. The Security Management Server stores the contract file and downloads it to Security Gateways during the upgrade. By verifying your status with the User Center, the contract file enables you to easily remain compliant with current Check Point licensing standards.

The license upgrade procedure can be performed if you have purchased any of the Enterprise Software Subscription services. To upgrade a VSX license, do the Software Blades upgrade procedure. See sk65850 http://supportcontent.checkpoint.com/solutions?id=sk65850.
The Trial Period

When installing a Check Point product for the first time, users receive a 15 day trial period, during which the product is fully functional. Once the trial period expires, you must purchase and install the appropriate permanent product licenses. During the trial period, you can define up to 25 Virtual Systems.

For More Information

For more information regarding licensing, refer to the Check Point User Center http://usercenter.checkpoint.com.
Using VSX with Multi-Domain Security Management

In This Section:

Overview ........................................................................................................................ 71
VSX Provisioning ........................................................................................................... 72
Defining Multi-Domain Servers ................................................................................... 73
Working with Virtual Devices ....................................................................................... 73

You can manage a VSX deployment using Multi-Domain Security Management. This chapter assumes that you are familiar with the Multi-Domain Security Management product. Only procedures specific to VSX deployments are discussed.

Overview

Check Point Multi-Domain Security Management is a centralized security management solution that addresses the unique requirements of service providers and large enterprises. By using Multi-Domain Security Management, administrators can centrally manage multiple independent networks, often belonging to different Domains, divisions, or branches.
<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SmartDomain Manager</td>
</tr>
<tr>
<td>2</td>
<td>Multi-Domain Server</td>
</tr>
<tr>
<td>3</td>
<td>SmartDashboard</td>
</tr>
<tr>
<td>4</td>
<td>Domain Management Server</td>
</tr>
<tr>
<td>5</td>
<td>Main Domain Management Server</td>
</tr>
<tr>
<td>6</td>
<td>VSX Gateway</td>
</tr>
<tr>
<td>7</td>
<td>Virtual Systems in Domain Management Servers</td>
</tr>
</tbody>
</table>

The **Multi-Domain Server** is a central management server that hosts the network management and security policy databases for these networks. Each independent domain is represented by a **Domain**, which provides the full functionality of a Security Gateway. Each Domain Management Server can host Virtual Systems, Virtual Routers and Virtual Switches as well as physical Check Point Gateways.

The Domain Management Server that manages a VSX Gateway or cluster is known as a **Main Domain Management Server**. You can host multiple Gateways and/or clusters on one Multi-Domain Server. Virtual Systems belonging to a given Domain can be distributed among multiple VSX Gateways and clusters.

The **SmartDomain Manager** is a centralized management solution for Domains, Domain Management Servers and the Multi-Domain Security Management environment. Each Domain Management Server uses its own instance of SmartDashboard, which is accessible only via the SmartDomain Manager, to provision its virtual devices and physical Gateways, as well as to manage their security policies.

### VSX Provisioning

The procedures for provisioning and configuring VSX Gateways, clusters and virtual devices using the Multi-Domain Security Management model are essentially the same as described for the Security Gateway management model. The principle difference is that you must first create and configure each Domain and its associated Domain Management Server objects using the SmartDomain Manager.

Each individual Domain Management Server is functionally equivalent to one Security Gateway. It has its own SmartDashboard instance that you use to provision, configure and manage network objects and security policies.

The steps for provisioning a VSX environment in using the Multi-Domain Security Management model are as follows:

1. Define and configure Multi-Domain Server and Multi-Domain Log Server as appropriate for your deployment.
2. Create and configure a Domain and a main Domain Management Server for each VSX Gateway and/or VSX cluster using the SmartDomain Manager.
3. Create and configure VSX Gateway ("Creating a New VSX Gateway" on page 34) and/or cluster objects ("Creating a New Cluster" on page 85) using the main Domain Management Server SmartDashboard. Modify the default security policy for these objects if desired.

4. Define individual Domains and Domain Management Servers as required for your deployment.

5. Create and configure Virtual Systems ("Creating a New Virtual System" on page 43) and other virtual devices for each Domain using that Domain’s SmartDashboard.

Defining Multi-Domain Servers

This section briefly presents the procedures for installing and deploying Multi-Domain Server machines in a VSX / Multi-Domain Security Management environment.

When working with management High Availability, you define at least two Multi-Domain Server machines. You can also employ multiple Multi-Domain Server machines to efficiently distribute management traffic (management Load Sharing) by creating multiple Domain Management Servers for individual Domains. For Load Sharing, define a Domain Management Server for each Multi-Domain Server.

Working with Virtual Devices

When defining and managing virtual devices in Multi-Domain Security Management, you must use the SmartDashboard associated with a specific Domain Management Server. Otherwise, the configuration procedures are identical to those for a Security Gateway management model. Multi-Domain Security Management treats virtual devices much in the same manner as physical devices.

You can add as many Virtual Systems to Domain Management Servers as your license permits. Virtual Systems added to a Domain Management Server do not have to reside on the same VSX Gateway or cluster.

Adding Virtual System to a Domain Management Server

To add a new Virtual System to a Domain Management Server:

1. In the SmartDomain Manager, launch SmartDashboard from the appropriate Domain Management Server.
2. Create and configure the Virtual System ("Creating a New Virtual System" on page 43).
3. Define and install a security policy.

Adding Virtual Routers and Virtual Switches to a Domain Management Server

To add Virtual Routers and Virtual Switches to a Domain Management Server:

1. In the SmartDomain Manager, launch SmartDashboard from the appropriate Domain Management Server.
2. Create and configure Virtual Routers ("Creating a New Virtual Router" on page 52) and Virtual Switches ("Creating a New Virtual Switch" on page 50) as required.
Introduction to VSX Clusters

In This Section:

- VSX Clustering Overview .............................................................................................. 74
- Planning a Cluster Deployment ................................................................................... 75
- VSX High Availability ..................................................................................................... 77
- Virtual System Load Sharing (VSLS) ............................................................................ 77
- Bridge Mode .................................................................................................................. 81
- Using Virtual Switches in a Cluster ............................................................................. 83

This chapter presents a conceptual overview of VSX cluster deployments, with emphasis on clustering features and their application. This discussion assumes that the reader is familiar with network cluster applications and environments, particularly ClusterXL.

The Cluster Management chapter ("Working with VSX Clusters" on page 85) provides detailed configuration procedures, including instructions for enabling and using all VSX clustering features.

VSX Clustering Overview

VSX clusters provide redundancy and load sharing features for Virtual Systems and other virtual devices. A VSX cluster consists of two or more identical, interconnected VSX Gateways that ensure continuous data synchronization.

VSX High Availability ensures continuous operation by means of transparent gateway failover. Virtual System Load Sharing (VSLS) enhances system performance by distributing active Virtual Systems amongst cluster members.

The advantages of using clusters in a VSX environment include:

- Transparent failover in case of gateway or Virtual System failure
- State synchronization ensures zero downtime for mission-critical environments
- Load sharing maintains system throughput during peak demand
- Enhanced scalability for future traffic growth

Physical Clusters

VSX clustering is based on Check Point ClusterXL concepts. This section reviews these concepts, and then demonstrates how these principles apply to VSX virtualization.

In typical Security Gateway deployment, a cluster consists of two or more identical, interconnected physical Security Gateways that provide redundancy and/or Load Sharing. This cluster behaves as a single Security Gateway and is assigned its own IP address, which is known as its cluster IP or virtual IP. This cluster IP address is distinct from the physical IP addresses of its cluster members, which are hidden from the networks connected to the cluster.

Traffic from external networks or the Internet directed to the internal networks arrives at the external cluster IP address. Depending on the clustering mode (High Availability or Load Sharing),
a designated cluster member receives the traffic and performs the required inspection. After inspection, traffic is either sent to its destination on the internal network, or dropped.

Internal networks send traffic destined for the Internet or external networks, to the cluster IP address. This traffic is processed by the designated cluster member, inspected, and forwarded to its external destination.

Each member interface has a unique, physical IP addresses. These IP addresses which are invisible to physical networks, are used for internal communication between members and the management server for such tasks as downloading policies, sending logs and checking the status of individual cluster members.

**VSX Clusters**

VSX clusters, like their physical counterparts, connect two or more synchronized Gateways in such a way that if one fails, another immediately takes its place. VSX clusters are defined at two levels:

VSX ensures that Virtual Systems, Virtual Routers, Virtual Switches and their interfaces are provisioned and configured identically on each cluster member. The figure below shows that each cluster member contains identical instances of each virtual device. These identical instances are referred to as **peers**.

VSX provides the management functionality to support network and security virtualization, including:

- **Assigning virtual IP addresses**: Each virtual device interface requires its own virtual IP address.
- **State synchronization**: virtual device state tables are synchronized to peers on other cluster members.

**Planning a Cluster Deployment**

As with physical network deployments, advance planning is the key to successfully creating a working network. IP address allocation for a VSX deployment requires particular attention. This section takes you through the basics of IP address allocation for a VSX environment. Your VSX configuration choices affect the number of IP addresses required, both public and private.
VSX Cluster Architecture

VSX IP address allocation is similar to physical networks. Both real and virtual IP addresses are required for network connectivity (internal and external), management, and state synchronization.

VSX simplifies the IP address management task by automatically assigning IP addresses to Warp Links between virtual devices. For example, Warp Links between a Virtual Router and its associated Virtual Systems are created automatically and assigned IP addresses without user intervention.

A VSX cluster network has these components:

- Synchronization Network
- Internal Communications Network
- Virtual IP addresses

**Synchronization Network**

The synchronization network is a physical network that carries state synchronization data between cluster members. You configure the synchronization network during the initial VSX cluster definition and can make changes as necessary when adding or removing members.

State Synchronization can be used ClusterXL deployments as well as other OPSEC-certified VSX solutions. The synchronization network must be configured using unique IP addresses that are not used anywhere else in the enterprise network.

**Internal Communication Network**

The internal communication network is a virtual network that is required for Check Point ClusterXL environments, in addition to the synchronization network. The internal communication network is invisible to external networks and lets cluster members communicate and recognize the state of the environment.

VSX assigns an IP address to the internal communication network during the cluster creation process. This eliminates the need to manually assign an IP address to each cluster member. The default IP address range consists of four class C networks:

IP address: 192.168.2.0
Net mask: 255.255.252.0

You can modify the default IP address using the Gateway Cluster Properties > Cluster Members page of the VSX cluster object, but only before creating Virtual Systems. Once Virtual Systems have been created, the IP range of the internal communication network cannot be modified.

**Note** - To avoid overlapping IP addresses, before creating any virtual devices, make sure the default IP address range of the Internal Communication network is not used anywhere else in the external network.

**Virtual IP Addresses**

Cluster (virtual) IP addresses are the only IP addresses visible to the external network. The assigned cluster IP addresses must correspond to the directly-connected subnet and server as a valid next hop address. These IP addresses are similar to virtual addresses configured across traditional cluster setups.
VSX High Availability

This section describes VSX High Availability features. In a VSX environment, you can work with one of two High Availability scenarios:

**VSX Gateway High Availability:** Each cluster member functions as a VSX Gateway and is synchronized with the other members. If one member goes down, it immediately fails over to another member. Likewise, if an individual Virtual System, Virtual Router or Virtual Switch goes down, the entire member fails over to another member. All members and Virtual Systems function in an active/active mode and are continuously synchronized.

**VSX Gateway High Availability**

VSX Gateway High Availability is the default cluster configuration. All members of a cluster must be configured to use the same clustering mode.

In the above example, member M1 experiences a failure the affects VS1 and all Virtual Systems immediately fail over to member M2.

**Virtual System Load Sharing (VSLS)**

Virtual System Load Sharing is a cluster technology that assigns traffic for Virtual Systems to different active cluster members, which has these benefits:

- **Capacity:** VSLS leverages the cluster machines to handle greater network volume by efficiently dividing the load.
- **Redundancy:** VSLS gives full redundancy by maintaining connectivity for all Virtual Systems even when individual members fail.
- **Scalability:** VSLS provides linear scalability for throughput and session rate.
- **Cost Effectiveness:** A VSLS cluster uses standard network switches to achieve cost effective Load Sharing.
- **Ease of Configuration:** Virtual Systems are automatically distributed among all the cluster members - no special configuration is required.
- **Priority Designation:** Mission-critical Virtual Systems can be separated from the other Virtual Systems, providing advantages in terms of bandwidth and resources.
- **System Scalability:** Every cluster member added to the cluster increases the overall system capacity and redundancy.

*Note* - These virtual devices are not supported when the Per Virtual System state is enabled:
- Virtual Routers
- Virtual Switches without physical or VLAN interfaces
Requirements

- VSLS requires Check Point ClusterXL.
- VSLS requires that all Virtual Systems in all cluster members have direct connectivity with each other. Connectivity must be accomplished using switches or VLAN connections. This is required for detecting and assigning Virtual System states.
- VSLS does not support Virtual Routers.

Conceptual Overview

This section presents a detailed conceptual overview of ClusterXL Virtual System Load Sharing.

Introduction

Virtual System Load Sharing is a cluster technology that assigns traffic for Virtual Systems to different active cluster members. This methodology is different from physical cluster load sharing because it is not connection-based. Virtual System Load Sharing is useful for mission-critical deployments, where reserving bandwidth for a particular Virtual System is a priority.

VSLS lets administrators manually assign Virtual Systems to specified cluster members, or lets VSX automatically assign Virtual System traffic dynamically.

Note - You cannot configure a VSX ClusterXL in the Load Sharing mode if the cluster contains Virtual Routers.

Virtual System Priority

Virtual System priority refers to a preference regarding which member hosts a Virtual System’s active, standby, and backup states. This preference is expressed as an integer value.

<table>
<thead>
<tr>
<th>Priority</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Highest priority, indicating the cluster designated to host the Virtual System active state.</td>
</tr>
<tr>
<td>1</td>
<td>Second highest priority, indicating the member designated to host the Virtual System standby state.</td>
</tr>
<tr>
<td>&gt; 1</td>
<td>Lower priorities, indicating the members designated to host a Virtual System backup state. The cluster member assigned priority 2 will be the first to switch the Virtual System to the Standby state in the event of a failure of either the Active or Standby Virtual System. A cluster member assigned priority 3 would be the next in line to come online in the event of another failure.</td>
</tr>
</tbody>
</table>

You can change the priority designation (“Distributing Virtual Systems Amongst Members” on page 101) using the vsx_util vsls command.

Virtual System Weight

Since all Virtual Systems are not equal in terms of traffic and load, VSLS allows you to assign “weights” to individual Virtual Systems. The weight of a Virtual System affects the dispersal pattern of other Virtual Systems across cluster members. Assigning a heavier weight to a Virtual
System gives it a larger share of a particular member’s resources, and accordingly, disperses the other Virtual Systems to other cluster members.

By default, all Virtual Systems are assigned an equal weight factor of 10. You can change the weight factor (“Distributing Virtual Systems Amongst Members” on page 101) using the `vsx_util vsls` command.

**Virtual System States**

VSLS adds a backup state to the existing active and standby states. The backup state contains the latest configuration settings for each Virtual System, but does not receive state table synchronization. The relationship between Virtual System states is illustrated in the below figure.

Each Virtual System peer in a VSLS cluster is replicated on all cluster members, and each copy exists in a different state. The active and standby states are synchronized so that the standby peer can immediately become active in the event of a failure of the active Virtual System or member. When this happens, the backup peer becomes the standby, and immediately synchronizes with the new active Virtual System.

VSLS reduces the load on the synchronization network by not synchronizing the backup Virtual System state tables with the active Virtual System until a failover occurs.

**Normalized VSLS Deployment Scenario**

For example, you can have three cluster members, each with three Virtual Systems. In this configuration, an equalized Load Sharing deployment could have one active Virtual System on each cluster member.
A different cluster member can host the Active state of each Virtual System. This distribution of Virtual Systems spreads the load among the clustered machines. When a Virtual System is created, the system automatically creates Standby and Backup states and distributes them among the other cluster members.

**Member Failure Scenario**

In the event that a member fails or experiences a connectivity problem, VSLS detects the problem and routes traffic for the affected Virtual Systems to their respective standby Virtual Systems. Standby Virtual Systems, which are fully synchronized with their active peers, change immediately to the active state and preserve active connections. At the same time, the backup Virtual Systems switch to standby, and synchronize fully with the newly active Virtual Systems.

In this scenario, Member 1 fails and its active and standby Virtual Systems fail over to Members 2 and 3. The active Virtual System (VS1) moves to Member 2 and directs all VS 1 traffic itself. Its backup peer on Member 3 synchronizes with the new active Virtual System and becomes the standby.

VS2 on Member 2 becomes the standby and synchronizes with the active peer on Member 3. For VS3, the active and standby peers remain the same.
Virtual System Failure Scenario

In a failure scenario where an active Virtual System fails on one member, but the standby and backup Virtual Systems remain up: the active Virtual System fails over to its standby peer, and its backup becomes the standby. The new standby synchronizes with the new active member.

All other Virtual Systems continue to function normally and no failover occurs.

Failure Recovery

When the failed cluster member or Virtual System comes back online, the system returns to its original Load Sharing configuration.

Bridge Mode

By implementing native layer-2 bridging instead of IP routing, you can add Virtual Systems without adversely affecting the existing IP structure.

When in the Bridge mode, Virtual System interfaces do not require IP addresses. You can optionally assign an IP address to the Virtual System itself (not the interfaces) to enable layer-3 monitoring, which provides network fault detection functionality.

VSX supports these Bridge mode models:

- **Active/Active (STP) Bridge Mode**: Provides redundancy while preventing undesirable loops between redundant switches.

- **Active/Standby Bridge Mode**: Provides path redundancy and loop prevention, while offering seamless support for Virtual System Load Sharing and overcoming many of the limitations of STP.

Spanning Tree Protocol (STP) Bridge Mode

The Spanning Tree Protocol is an industry standard technology to prevent loops in high-speed switched networks. To use the STP Bridge mode, you must have STP deployed and properly configured on your network. These STP layer-2 protocols are supported:
• 802.1q
• 802.1D
• 802.1s
• 802.1w
• PVST+

See your vendor documentation to learn how to deploy and configure STP on your network hardware.

Active/Standby Bridge Mode

The Active/Standby Bridge Mode enhances both High Availability [for significant improvements] and Virtual System Load Sharing in VSX clustered environments (VSLS) [for throughput distributed among Virtual Systems].

Active/Standby Bridge Mode has these advantages:

• Instantaneous failover
• Enhanced administrator control over bridge failover.
• VSLS support
• VLAN translation

The principal limitation of the Active/Standby Bridge Mode is that it breaks the STP tree structure.

Note - When configuring a Virtual System in the Active/Standby Bridge Mode, you should remove Virtual System VLANs from the STP database in the switches. This action prevents delays due to trunk interface failback.

Deployment Scenarios

This section presents illustrative Active/Standby Bridge Mode deployments, which cannot function using a standard STP Bridge mode configuration.
**VLAN Shared Interface Deployment**

In this deployment, each member connects to a pair of redundant switches through a VLAN trunk. All Virtual Systems in a given member share the same VLAN trunk.

With Active/Standby Bridge Mode in High Availability mode, ClusterXL directs traffic to members according to administrator-defined priorities and status. In Virtual System Load Sharing deployments, the system distributes the traffic load amongst members according to your Virtual System Load Sharing configuration.

**Three Layer Hierarchical Model**

A three-layer hierarchical model is used in large, high-traffic network environments.

1. A **core network**, with high-speed backbone switches that direct traffic to and from the Internet and other external networks.
2. A **distribution layer**, with routers, for connectivity between the core and the access layer.
3. An **access layer**, with redundant LAN switches, that forward traffic to and from internal networks.

VSX in Active/Standby Bridge Mode is incorporated in the **distribution** layer, enforcing the security policy.

The routers direct external traffic to the appropriate Virtual System through a segregated VLAN. Inspected traffic exits the Virtual System through a separate segregated VLAN, to the routers and then to internal destinations.
Using Virtual Switches in a Cluster

In a VSX cluster, Virtual Switches are also clustered for redundancy. Virtual Switches in the
cluster are defined as active/active.

By means of the ClusterXL Control Protocol (CCP), the physical interface connected to the Virtual
Switch is monitored. In the event of a failover, all Virtual Systems on standby become active, and
send gratuitous ARPs from the warp interface between the Virtual System and the Virtual Switch.

In the above figure, a simplified VSX cluster contains two members, one active, and the other
standby. The Virtual Switches within each cluster are active/active. When the physical interface
connected to either Virtual Switch fails to respond, a failover occurs.
Working with VSX Clusters

In This Section:

- Configuration Overview .......................................................... 85
- Creating VSX Clusters .............................................................. 85
- Modifying a Cluster Definition .................................................. 88
- Working with Cluster Members .................................................. 93
- Changing the Cluster Type ....................................................... 96
- Enabling VSX Gateway High Availability ................................. 99
- Configuring Virtual System Load Sharing ............................... 99
- Configuring Virtual Systems in Bridge Mode ......................... 106
- Advanced Clustering Configuration ........................................ 108

This chapter presents the procedures for configuring VSX in various cluster deployment scenarios. In addition to the basic scenarios, conceptual material and illustrative examples are presented for several advanced features, including the Bridge mode and dynamic routing.

You will use SmartDashboard for most of the basic cluster configurations. You will need the command line interface to add more members, remove members, and upgrade members. Many advanced cluster management procedures require the command line.

Configuration Overview

The majority of the basic cluster configuration process is performed using SmartDashboard, both in Security Management and Multi-Domain Security Management models. However, you will need to use the command line interface to add additional members, remove members and upgrade existing members to VSX clusters. Many advanced cluster management, including Load Sharing definitions, require the command line.

Creating VSX Clusters

Creating a New Cluster

This section describes how to create a new VSX cluster using the VSX Cluster Wizard. The wizard guides you through the following steps to configure a VSX cluster.

After completing the VSX Cluster Wizard, you can modify most cluster and member properties directly from SmartDashboard.

To create a new cluster:

1. Open SmartDashboard.
   If you are using Multi-Domain Security Management, open SmartDashboard from the Domain Management Server in which you are creating the cluster.

2. From the Network Objects tree, right-click Check Point and select VSX > Cluster.
   The General Properties page of the VSX Cluster Wizard opens.
Defining Cluster General Properties

The Cluster General Properties page contains basic identification properties for VSX clusters. This window contains the following properties:

- **VSX Cluster Name**: Unique, alphanumeric name for the cluster. The name cannot contain spaces or special characters except the underscore.
- **VSX Cluster IP Address**: Management interface IP address.
- **VSX Cluster Version**: VSX version to use for this cluster.
- **VSX Cluster Platform**: Platform type hosting the cluster members.
  - To create a HA cluster, select **Check Point SecurePlatform (ClusterXL)** from the list.
  - To create a Load Sharing (VSLS) cluster, select **Check Point ClusterXL Virtual System Load Sharing** from the list.

**Note** - All cluster members must use the type of platform, with the same specifications and configuration.

Selecting Creation Templates

The Virtual Systems Creation Templates allows you to select a Virtual System Creation Template that automatically applies predefined, default topology and routing definitions to Virtual Systems when they are first created. This feature ensures consistency among Virtual Systems and speeds up the provisioning process.

You always have the option of overriding the default creation template when creating or modifying a Virtual System.

The available creation templates are as follows:

- **Shared Interface**: All Virtual Systems share a single external interface, but maintain separate internal interfaces.
- **Separate Interfaces**: All Virtual Systems use their own separate internal and external interfaces. This template creates a Dedicated Management Interface (DMI) by default.
- **Custom Configuration**: You manually create a custom configuration without any template.

Adding Members

The VSX Cluster Members window defines the members of the new cluster. You must define at least two cluster members, and up to as many as eight members. You can add new members later.

To add a new cluster member:

1. In the **VSX Cluster Members** window, click **Add**.
2. The **Member Properties** window opens.
3. Enter the name and its IP address for the cluster member.
4. Enter and confirm the activation key to initialize SIC trust between the cluster member and the management server.
5. Do these steps again for all the cluster members.
Defining Cluster Interfaces

The **VSX Cluster Interfaces** window lets you define physical interfaces as VLAN trunks. The list displayed contains all interfaces currently defined on the gateway machine or cluster.

To configure a VLAN trunk:

Select an interface to define it as a VLAN trunk. You can clear an interface to remove the VLAN trunk assignment.

**Important** - You cannot define the management interface as a VLAN trunk. To use a VLAN as the management interface, you must define the VLAN on the Security Gateway before you use SmartDashboard to create the VSX Gateway.

Configuring Cluster Members

If you selected the custom configuration option, the **VSX Cluster Members** window appears. In this window, you define the synchronization IP address for each member.

To configure the cluster members:

1. Select the synchronization interface from the list.
2. Enter the synchronization interface IP address and net mask for each member.

To use a VLAN as a synchronization interface:

1. Define the VLAN on the Security Gateway.
2. Open SmartDashboard and create the VSX Gateway.
3. On the VSX Gateway, from the CLI open `fwkern.conf` and add this line:
   
   ```
   fwha_monitor_all_vlan=1
   ```

Cluster Management

The **VSX Gateway Management** page allows you to define several security policy rules that protect the cluster itself. This policy is installed automatically on the new VSX cluster.

**Note** - This policy applies only to traffic destined for the cluster. Traffic destined for Virtual Systems, other virtual devices, external networks, and internal networks is not affected by this policy.

The security policy consists of predefined rules covering the following services:

- **UDP**: SNMP requests
- **TCP**: SSH traffic
- **ICMP**: Echo-request (ping)
- **TCP**: HTTPS (secure HTTP) traffic

Configuring the Cluster Security Policy

1. **Allow**: Enable a rule to allow traffic for those services for which you wish to allow traffic. Clear a rule to block traffic. By default, all services are blocked.
   
   For example, you may wish to allow UDP echo-request traffic in order to be able to ping cluster members from the management server.

2. **Source**: Click the arrow and select a **Source Object** from the list. The default value is *Any.*
Click New Source Object to define a new source.

For more about security policies, see the R76 Security Management Administration Guide http://supportcontent.checkpoint.com/solutions?id=sk105938.

Completing the Wizard

To complete the VSX Cluster Wizard:

1. Click Next to continue and then click Finish to complete the VSX Cluster wizard.

   It can take several minutes to complete. A message appears indicating successful or unsuccessful completion of the process.

   If the process ends unsuccessfully, click View Report to view the error messages. See to the troubleshooting steps (“VSX Diagnostics and Troubleshooting” on page 157) for more information

2. In SmartConsole, double-click the new VSX Cluster object.

3. Click ClusterXL and make sure that the Use State Synchronization option is enabled.

Converting a Security Gateway Cluster to VSX

Use the VSX Gateway Conversion wizard in SmartDashboard to convert a Gaia High Availability cluster of Security Gateways to a VSX cluster (“Converting Gateways to VSX Gateways” on page 37). The settings of each Security Gateway are applied to the VSX Gateway (VS0). For more about using the Conversion wizard, see sk79260 http://supportcontent.checkpoint.com/solutions?id=sk79260.

You can only convert a cluster that uses the Gaia operating system.

Important - There is no loss of connectivity during the conversion process. You cannot use the conversion wizard to convert a Load Sharing cluster of Security Gateways.

Modifying a Cluster Definition

Once you create a cluster using the wizard, you can modify the topology and other parameters using the VSX Cluster Properties window. This window also allows you to configure many advanced features not available with the wizard.

To work with a VSX cluster definition, double-click on the cluster object in the SmartDashboard Network Object tree. The VSX Cluster Properties window opens, showing the General Properties page.

Most cluster objects and properties can be defined using the SmartDashboard GUI. Several definitions, however, require CLI commands, while others may be performed using either method.

A brief explanation for each of the definition pages follows. More detailed explanations for features that are not specific to VSX (NAT, IPS, VPN, etc.) are available in the online help or in the appropriate product Administration Guide.

General Properties

See the General Properties page to view general properties and to activate Check Point products for use with this cluster and its members.
You can modify the following properties:

- **Comment**: Free text comment that appears in the Object List and elsewhere
- **Color**: Color of the object icon as it appears in the Network Object tree
- **Network Security**: Select Check Point products and Software Blades active on this cluster and its members

**Cluster Members**

The **Cluster Members** page lets you view and modify several properties for individual cluster members, including IP addresses for members and the internal communication network. You can also view where cluster and member objects in the object database are used.

**Gateway Cluster Member List**

The **Cluster Members** page shows all the VSX cluster members on the VSX Gateway.

To edit a cluster member:

From the **Cluster Member** page, select a member and click **Edit**.

The **Cluster Member Properties** window opens. These are the settings that you can edit:

- **General Tab**:
  - **Comment**: Free text comment that appears in the Object List and elsewhere
  - **Color**: Color of the object icon as it appears in the Object Tree
  - **Secure Internal Communication**: Check and reset SIC trust

- **Topology Tab**: Displays the member IP address and net mask for each interface. Double click on an interface to displays its properties.

- **NAT Tab**: Define NAT rules for cluster members connected to a Virtual Router.

- **VPN Tab**: Contains a variety of configuration properties for site-to-site VPN deployments. This window is only available if the Check Point VPN product is enabled on the **General Properties** page.

**Where Used**

Click **Where used** to show information about the selected member in the objects database.

- **Name**: Cluster name.
- **Table**: Name of the table in the database under which the selected object is listed.
- **Is removable**: Specifies whether or not you are allowed to remove the selected object. If the object is not removable and nevertheless you choose to remove it, it will impact the database or Rule Base.
- **Refresh**: Click to update the window display if you make changes.
- **Context**: Where the object is used.

**Internal IP Address and Net Mask**

VSX creates an internal communication network and automatically assigns it an IP address and net mask from a predefined pool. You can change this IP address here if you have not yet defined a
Virtual System. Although traffic from this address is never sent to any networks, you must ensure that this IP address is unique and not in use anywhere on your defined network.

**ClusterXL**

You can enable or disable state synchronization in the ClusterXL window and choose options to track changes in the state of cluster members on this page. All other properties are ClusterXL configuration properties are disabled. You can modify the ClusterXL configuration using the `vsx_util` command.

**Creation Templates**

The Creation Templates page displays the creation template used to create Virtual Systems. You can change from the current creation template to the Custom Configuration template and change the shared physical interface if the Shared Interface template is active.

- Select the Custom Configuration option to change from the Shared Interfaces or Separate Interfaces templates.
  - You cannot change back from the Custom Configuration template once you have completed the definition and saved it to the configuration to cluster.
- To change the shared interface, click Settings and select an interface from the list that appears.

**Physical Interfaces**

The Physical Interfaces page allows you to add or delete a physical interface on the VSX Gateway, and to define interfaces to be used as VLAN trunks.

- To add a new physical interface, click Add and enter the interface name in the appropriate field.
- To define an interface as a VLAN trunk, select the desired interface and enable the VLAN Trunk option. To disable a VLAN trunk, clear the option.

**Synchronization**

The Synchronization window displays the state synchronization network. There are no configurable properties.

**Topology**

The Topology page contains interface and routing definitions.

**Interfaces**

The Interfaces section defines interfaces and links to devices. You can add new interfaces as well as delete and modify existing interfaces.

**To add an interface:**

1. Click New and select one of these options:
   - Regular - Create a new interface
   - State Synchronization
• Leads to Virtual Router
• Leads to Virtual Switch

The Interface Properties window opens. Click Actions > Copy to Clipboard to copy the Interfaces table in CSV format.

2. Define the appropriate properties ("Working with Interface Definitions" on page 56).
3. Click OK.

To change an interface:
1. Double-click an interface.
   The Interface Properties window opens.
2. Change the parameters for the interface ("Working with Interface Definitions" on page 56).
3. Click OK.

To delete an interface:
1. From the Topology page, select the interface and click Delete.
2. Click OK.

Routes

The Routes section of the Topology window defines routes between network devices, network addresses, and virtual devices. Some routes are defined automatically based on the interface definitions. You can add new routes as well as delete and change existing routes.

To add a default route to the routing table:
1. Click Add Default Route.
   The Default Gateway window opens.
2. Enter the default route IP address or select the default Virtual Router.
3. Click OK.
   The default route is added to the routing table.
4. Select the default route and click Edit.
   The Route Configuration window opens.
5. Configure the settings for the default route and click OK.

To add a new route to the routing table:
1. Click Add.
   The Route Configuration window opens.
2. Configure the Destination IP address and netmask.
3. Configure the next hop IP address or Virtual Router.
4. Optional: Select Propagate route to adjacent virtual devices to “advertise” the route to neighboring virtual devices, and enable connectivity between them.
5. Click OK.

To change a route:
1. Select the route.
2. Click Edit.
The Route Configuration window opens.
3. Change the settings.
4. Click OK.

To delete a route:
1. Select the route.
2. Click Remove.
A confirmation window opens.
3. Click OK.

Calculating Topology Automatically Based on Routing Information
Enable this option to allow VSX to automatically calculate the network topology based on interface and routing definitions (enabled by default). VSX creates automatic links, or connectivity cloud objects linked to existing internal or external networks.
- This option is not available in Bridge mode.
- When employing dynamic routing, it is recommended to disable this option.

VPN Domain
The VPN Domain section in the Topology page defines the set of hosts that use a VPN tunnel to communicate with peer Virtual Systems.

Define a VPN Domain to include a virtual device as part of the VPN connection. The domain defines the Virtual System interfaces that are in the VPN. You can define a VPN Domain in different ways:
- All IP Addresses behind Cluster Members are based on topology information: Includes all hosts not located behind an external gateway cluster interface.
- Manually Defined: Includes all hosts in the selected network or group.
- Remote Access Communities: Define an alternative VPN domain for Remote Access Community traffic.

To specify the VPN domain:
1. Click Set domain for Remote Access Community.
The VPN Domain per Remote Access Community window opens.
2. Double-click a Remote Access Community.
The Set VPN Domain window opens.
3. Select a VPN domain from the list, or click New, to define a new domain.
4. Click OK.

NAT
The NAT > Advanced page lets you configure NAT rules for packets originating from a Virtual System.

To enable and configure NAT for a Virtual System:
1. Select Add Automatic Address Translation.
2. Select a translation method:
• **Hide**: Hide NAT only allows connections originating from the internal network. Internal hosts can access internal destinations, the Internet and other external networks. External sources cannot initiate a connection to internal network addresses.

• **Static**: Static NAT translates each private address to a corresponding public address.

3. If you select **Hide**, select one of these options:
   • **Hide behind Gateway** hides the real IP address behind the Virtual System external interface IP address,
   or
   • **Hide behind IP Address** hides the real address behind a virtual IP address, which is a routable, public IP address that does not belongs to any real machine.

4. If you selected **Static NAT**, enter the static IP address in the appropriate field.

5. Select the VSX Gateway from the **Install on Gateway** list.

**VSX Bridge Configuration**

The **VSX Bridge Configuration** page allows you to specify the loop detection algorithm when working in the Bridge mode.

Enable the **Check Point ClusterXL** option to enable the **Active/Standby Bridge Mode** loop detection algorithms contained in ClusterXL.

Enable the **Standard Layer-2 Loop Detection Protocols** to use standard loop detection protocols, such as STP or PVST+.

**Cooperative Enforcement**

Cooperative Enforcement works with Check Point Endpoint Security servers. This feature utilizes the Endpoint Security server compliance capability to verify connections arriving from various hosts across the internal network. The Cooperative Enforcement window contains several configuration properties for defining this feature. For more information, please refer to the online help and the **R76 Firewall Administration Guide**


**Changing the Cluster Management IP and/or Subnet**

You can change the cluster management IP address and/or subnet by executing the `vsx_util change_mgmt_ip` and `vsx_util change_mgmt_subnet` commands.

**Changing the Internal Communication Network IP**

You can change the internal communication network IP address by using the `vsx_util change_private_net` command.

**Working with Cluster Members**

This section presents procedures for adding and deleting cluster members, as well as for upgrading existing cluster members to VSX.
Adding a New Member

⚠️ **Important** - Verify that no other administrators are connected to the management server before proceeding. The vsx_util command cannot modify the management database if the database is locked because other administrators are connected.

To add a new member to an existing cluster:

1. Close SmartDashboard and backup the management database.
2. From the management server CLI, enter expert mode.
3. Run the vsx_util add_member command and follow the on-screen instructions.
4. Wait until the **add member operation finished successfully** message appears, indicating that the database has been successfully updated and saved.
   
   **Note** - In a Multi-Domain Security Management environment, this operation will skip any Domain Management Servers locked by an administrator. If this should occur, run the operation again for the relevant Domain Management Servers once they become available.
5. Open SmartDashboard and verify that an object representing the new member appears in the specified cluster.
6. If necessary, modify the cluster configuration.
7. Close SmartDashboard.
8. From the management server CLI, enter expert mode.
9. Run the vsx_util add_member_reconf command and follow the on-screen instructions.
10. Wait until the **Reconfigure module operation completed successfully** summary notice appears.
    
    **Note** - In a Multi-Domain Security Management environment, the operation will skip any Domain Management Servers locked by an administrator. If this should occur, run the operation again for the relevant Domain Management Servers when they become available.
11. Reboot the new member.
12. If the cluster is running in the VSLS mode, run vsx_util vsls to redistribute Virtual Systems to the newly added member.

Deleting a Member

⚠️ **Important** - Verify that no other administrators are connected to the management server before proceeding. The vsx_util command cannot modify the management database if the database is locked.

You perform this operation using the management server command line. It is strongly recommended that you back up the database prior to removing a member.

To remove a member from a cluster:

1. Detach the license from the member to be removed. You cannot remove a member if the license is attached.
2. Close SmartDashboard.
3. From the management server command line, run the vsx_util remove_member command. Perform the following tasks as prompted:
   a) Enter the Security Gateway or main Domain Management Server IP address.
b) Enter the administrator name and password.

c) Type 'y' to confirm that you have detached the license from the member.

d) Enter the cluster name.

e) Enter the cluster member name.

f) Type 'y' to confirm that the member to be removed has been disconnected.

4. Wait until the remove member operation finished successfully message appears. The database is now updated and saved. In SmartDashboard, the object for the deleted member no longer appears in the specified cluster.

5. Open SmartDashboard and verify that deleted member no longer appears in the specified cluster.

Note - In a Multi-Domain Security Management environment, the operation will skip any Domain Management Servers locked by an administrator. If this should occur, run the operation again for the relevant Domain Management Servers when they become available.

Upgrading Cluster Members

This section describes the procedures for upgrading cluster members that were initially installed using an earlier version of VSX. You perform the upgrade process using the vsx_util upgrade command. Afterwards, you use the vsx_util reconfigure command to apply settings stored in the management database to the newly upgraded member.

Upgrading a Member to the Current Version

Important - Verify that no other administrators are connected to the management server before proceeding. The vsx_util command cannot modify the management database if the database is locked.

Performing the following steps to upgrade the cluster and its members:

1. Close SmartDashboard.
2. Enter the Expert mode.
3. Execute the vsx_util upgrade command from the management server command line.
   Enter the following information when prompted:
   a) Security Gateway or main Domain Management Server IP address
   b) Administrator name and password
   c) Cluster name
4. When prompted, select the version to which you wish to upgrade.
5. Wait until the Finished upgrading/database saved successfully message appears, indicating that the database has been updated and saved.
6. Open SmartDashboard and verify that an object representing the new member now appears in the specified cluster.

Note - In a Multi-Domain Security Management environment, the operation will skip any Domain Management Servers locked by an administrator. If this should occur, run the operation again for the relevant Domain Management Servers when they become available.
7. Perform a fresh installation of VSX on each upgraded member.
8. Perform the initial configuration steps on each member as described in the *R76 Installation and Upgrade Guide*
   
   http://supportcontent.checkpoint.com/documentation_download?ID=22901. These steps include:
   
   a) Define the IP address, net mask and default gateway.
   
   b) Install a valid license.
   
   c) Set the SIC activation key.
   
   d) Configure the cluster properties as required. These property settings must be the same as defined for the other cluster members.

9. Run the `vsx_util reconfigure` command from the management server command line. Enter the following information when prompted:
   
   a) Management server or main Domain Management Server IP address
   
   b) Administrator name and password
   
   c) SIC activation key for the upgraded member

   This action installs the existing security policy and configuration on the newly upgraded members.

10. Wait until the `Finished upgrading/database saved successfully` message appears.
    
    **Note** - In a Multi-Domain Security Management environment, the operation will skip any Domain Management Servers locked by an administrator. If this should occur, run the operation again for the relevant Domain Management Servers when they become available.

11. Reboot each member.

**Notes to the Upgrade Process**

- You only need to run the `vsx_util upgrade` command once for each VSX cluster. You must, however, run the `vsx_util reconfigure` command for each cluster member.

  For example, for a deployment with two clusters, each cluster having three members, run `vsx_util upgrade` twice, once for each cluster object, and the `vsx_util reconfigure` six times, once for each cluster member.

- To ensure stability of the VSX deployment, run the `vsx_util upgrade` and `vsx_util reconfigure` consecutively. For example, run the `vsx_util upgrade` command for the cluster object first and then run the `vsx_util reconfigure` for the first member. Run `vsx_util reconfigure` for the second and subsequent members immediately thereafter.

- Verify that each upgraded member is fully operational before upgrading the remaining members.

- You cannot install policies until the upgrade process has completed successfully for all members.

**Changing the Cluster Type**

This section presents procedures for converting cluster members from one cluster type (High Availability or VSLS) to the other. Changing the cluster mode involves the use of the `vsx_util convert_cluster` command.
Converting from VSLS to High Availability

Do these procedures to convert a cluster from VSLS to High Availability:
1. Redistributing all active Virtual Systems to one member
2. Disabling VSLS options
3. Converting the cluster to High Availability

**Redistributing Active Virtual Systems to One Member**

To redistribute all active Virtual Systems to one member:
1. Close SmartDashboard.
2. Enter the Expert mode.
3. Execute the `vsx_util vsls` command.
4. Enter the Security Management Server or Multi-Domain Security Management Domain Management Server IP address.
5. From the Load Sharing, enter “3. Set all VSs active on one member”.
6. Enter the administrator user name and password.
7. Enter the VSX cluster name.
8. Enter the number corresponding to the member designated to host all active members.
9. Enter “y” to save and apply the configuration.
10. Exit the Load Sharing menu.

When the `convert_cluster` command finishes, there should be only one active member on which all Virtual Systems are in the active state, and one standby member on which all virtual devices are in the standby state. Any additional members should be in standby mode and their virtual devices in the down state.

**Disabling VSLS Options**

To convert existing cluster members to the VSX Gateway High Availability mode:
1. On each member, execute the `cpconfig` command and do the following:
   a) Disable the Per Virtual System State for each member.
   b) Disable ClusterXL for Bridge Active/Standby for each member.
2. Re-initialize the members using the `cpstop` and `cpstart` commands.

**Converting the Cluster**

To convert the cluster to High Availability:
1. Execute the `vsx_util convert_cluster` command.
2. Enter the Security Management Server or Multi-Domain Security Management Domain Management Server IP address.
3. From the Load Sharing menu, enter “3. Set all VSs active on one member”.
4. Enter the administrator user name and password.
5. Enter the VSX cluster name.
6. Enter "HA"
7. Re-initialize all members using the `cpstop` and `cpstart` commands.

Converting from High Availability to VSLS

To convert an existing High Availability cluster to VSLS Load Sharing:
1. Close SmartDashboard.
2. On each member:
   a) Run `cpconfig`
   b) Enable the **Per Virtual System State**.
   c) Enable **ClusterXL for Bridge Active/Standby**.
3. Restart the members: `cpstop` and `cpstart`
4. On the management server, enter Expert mode.
5. Run: `vsx_util convert_cluster`
7. Enter the administrator user name and password.
8. Enter the VSX cluster name.
9. Enter: **LS**
10. At the "**Proceed with conversion?**" prompt, enter: **y**
11. Select an option to distribute Virtual Systems among members:
   a) Distribute all Virtual Systems equally.
   b) Set all Virtual Systems as **Active** on the same member.
12. Reboot the members.

**Note** - You cannot convert a VSX cluster to the VSLS mode if it contains Virtual Systems in the Active/Active Bridge mode or Virtual Routers.

Sample Command Output

The following screen printout shows an example of the output from the `vsx_util convert_cluster` command.
Enabling VSX Gateway High Availability

VSX Gateway High Availability is the default cluster configuration. If Load Sharing (VSLS) is not active, a cluster functions in the VSX Gateway High Availability mode. All members of a cluster must be configured to use the same clustering mode.

Configuring New Cluster Members

To configure members for VSX Gateway High Availability:

In the Gaia First Time Configuration Wizard Products page, select ClusterXL.

Configuring Virtual System Load Sharing

This section presents the various procedures for configuring VSLS deployments. You use the vsx_util vsls to perform various VSLS configurations tasks.

To start vsx_util vsls:

1. From the management server Expert mode, execute vsx_util vsls.
2. Enter the management server IP address.
3. Enter administrator user name and password.
4. Enter the VSX Gateway name.
5. From the VLSL menu, choose the desired option.

Enabling VLSL

In order to use VLSL for VSX, you must first activate the Per Virtual System State mode on each cluster member. You can then create a Load Sharing cluster, either by creating a new cluster object, or by converting an existing High Availability cluster to Load Sharing mode. After completing this process, you can modify Virtual Systems as required.

Enabling the Per Virtual System State Mode

The Per Virtual System State mode enables active Virtual Systems to be placed on different cluster members, and for Virtual System-specific failover. This setting is mandatory for VLSL. On each cluster member, do the following:

- **Note** - The following virtual devices are not supported when the Per Virtual System state is enabled:
  - Virtual Routers
  - Virtual Switches that do not have physical or VLAN interfaces

1. Run `cpconfig`.
2. Select Enable Check Point Per Virtual System State.
3. Answer y to the question: Would you like to enable Per Virtual System state?
4. Reboot the machine.
5. Repeat this procedure for each member.

Creating a New VLSL Cluster

To create a new VLSL cluster:

1. Open SmartDashboard.
2. From the Network Objects tree, right click Check Point and select VSX > Cluster.
   - The VSX Cluster Wizard opens.
3. Create and configure the new cluster ("Creating a New Cluster" on page 85).
   a) On the General Properties page, from VSX Cluster Platform, select Check Point ClusterXL Virtual System Load Sharing.
   b) On the Creation Templates page, select the creation template ("Selecting Creation Templates" on page 86).
   c) Complete the VSX Cluster Wizard.

Using the vsx_util vsls Command

You use the `vsx_util vsls` command to perform various Virtual System Load Sharing configuration tasks, including:

1. Displaying the current VLSL configuration
2. Distributing Virtual Systems equally amongst cluster members
3. Set all Virtual Systems as active on one member
4. Manually define the priority and weight for individual Virtual Systems
5. Import VSLS configurations from comma separated value (CSV) text files
6. Export VSLS configurations to comma separated value (CSV) text files
7. Exporting and Import VSLS configurations from/to comma separated value (CSV) text files

To work with the vsx_admin vsls command:
1. Run vsx_admin vsls from the Expert mode on the management server
2. Select the desired choice from the VSLS menu

vsx_admin vsls main menu

Enter Administrator Name: aa
Enter Administrator Password:
Enter VSX cluster object name: vsx

VS Load Sharing - Menu

1. Display current VS Load sharing configuration
2. Distribute all Virtual Systems so that each cluster member is equally loaded
3. Set all VSs active on one member
4. Manually set priority and weight
5. Import configuration from a file
6. Export configuration to a file
7. Exit

Enter redistribution option (1-7) [1]:

Distributing Virtual Systems Amongst Members

The primary advantage of VSLS is the ability to distribute active, standby and backup Virtual Systems amongst cluster members in order to maximize throughput and user response time. You can choose to distribute Virtual Systems according to one of the following options:

- Automatically distribute active Virtual Systems amongst cluster members so that all members are equally loaded based on assigned weights and existing or default priority definitions.
- Automatically place all active Virtual Systems on the same member
- Manually define priorities and weights for each Virtual System

Distributing Virtual Systems for Equal Member Loading

To distribute Virtual Systems for equal member loading:
1. From the VSLS menu, select "2. Distribute all Virtual Systems so that each cluster member is equally loaded".
2. At the "Save & apply configuration?" prompt, enter "y" to continue.

The process update process may take several minutes or longer to complete, depending on the quantity of Virtual Systems and cluster members.

Placing All Active Systems on the Same Member

1. From the VSLS menu, select "3. Set all VSs active on one member".
2. When prompted, enter the number corresponding to the member designated as the primary member.

3. When prompted, enter the number corresponding to the member designated as the standby member. All other members will be designated as backup members.

4. At the “Save & apply configuration?” prompt, enter "y" to continue.

The process update process may take several minutes or longer to complete, depending on the quantity of Virtual Systems and cluster members.

Assigning Priorities and Weights for a Single Virtual System

You can modify these settings in one of two ways:

- Automatically assign weights only to Virtual Systems. This method prompts you for a weight for each Virtual System and then automatically updates the settings.
- Manually assign both priorities and weights to individual Virtual Systems.

To automatically assign weights to all Virtual Systems:

1. From the VSLS menu, select Manually set priority and weight.

2. Enter "a" to automatically scroll through each Virtual System.

3. For each Virtual System, enter a weight value and press Enter.
   a) If you do not enter a weight value for a Virtual System, the currently assigned weight is retained.
   b) To stop entering weight values for additional Virtual Systems, enter s. Only those Virtual Systems that have been assigned a new weight value will be updated.

4. At the Save & apply configuration prompt, enter y to continue.

The process update process may take several minutes or longer to complete, depending on the quantity of Virtual Systems and cluster members.

To manually assign priorities and weights for individual Virtual Systems:

1. From the VSLS menu, select Manually set priority and weight.

2. Enter m to manually update both priorities and weights for individual Virtual Systems.

3. At the Would you like to change the Virtual System’s priority list? prompt, enter y to change the member priority.
   a) Enter the number associated with the member to receive the highest priority.
   b) Enter the number associated with the member to receive the next highest priority.
   c) Continue until all members have been assigned a priority.

4. At the Would you like to change the Virtual System’s weight? prompt, enter y to assign a weight n to retain the existing weight value.
   a) At the prompt, enter an integer between 1 and 100, representing the new weight value.

5. At the “Do you wish to configure another Virtual System?” prompt, enter “y” to configure another Virtual System or “n” to continue.

6. At the “Save & apply configuration?” prompt, enter “y” to continue.

The process update process may take several minutes or longer to complete, depending on the quantity of Virtual Systems and cluster members.
Viewing VSLS Status

To view the current VSLS status and Virtual System distribution amongst members, select “1. Display current VS Load Sharing configuration” from the VSLS menu. The output is similar to the below example:

<table>
<thead>
<tr>
<th>VSID</th>
<th>VS name</th>
<th>gw150</th>
<th>gw151</th>
<th>gw152</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>vs1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>vs2</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>vs3</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>vs5</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>vs4</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td><strong>Total weight</strong></td>
<td>20</td>
<td>20</td>
<td>10</td>
<td>50</td>
</tr>
</tbody>
</table>

Legend:
0 - Highest priority
1 - Next priority
2 - Lowest priority

Virtual System Priority

Virtual System priority refers to a preference regarding which member hosts a Virtual System’s active, standby, and backup states. This preference is expressed as an integer value.

<table>
<thead>
<tr>
<th>Priority</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Highest priority, indicating the member designated to host the Virtual System active state.</td>
</tr>
<tr>
<td>1</td>
<td>Second highest priority, indicating the member designated to host the Virtual System standby state.</td>
</tr>
<tr>
<td>&gt; 1</td>
<td>Lower priorities, indicating members designated to host a Virtual System’s backup state. The cluster member assigned priority 2 will be the first to switch the Virtual System to the Standby state in the event of a failure of either the Active or Standby Virtual System. A cluster member assigned priority 3 would be the next in line to come online in the event of another failure.</td>
</tr>
</tbody>
</table>

Virtual System Weight

Each Virtual System is assigned a weight factor, which indicates its traffic volume relative to the total traffic volume (the sum of all weight factors) on a given cluster member. VSX uses the weight factor to determine the most efficient distribution of Virtual Systems amongst cluster members. System resource allocation is not affected by the weight factor, nor does VSX take weight into consideration for any other purpose.

By default, all Virtual Systems are assigned an equal weight factor of 10.

Exporting and Importing VSLS Configurations

When working with large scale VSLS deployments consisting of many Virtual Systems, multiple cluster members, using the vsx_util command to perform configuration tasks can be quite time consuming. To allow administrators to efficiently configure such deployments, VSX supports...
uploading VSLS configuration files containing configuration information for all Virtual Systems directly to management servers and cluster members.

This capability offers the following advantages:

- Rapid VSLS configuration of large-scale deployments with many Virtual Systems and cluster members.
- Efficient migration and scalability for complex deployments
- External backup of VSLS configurations.

VSLS configuration files are comma separated value (CSV) files that are editable using a text editor or another applications, such as Microsoft Excel. You can use the configuration file to rapidly change the weight and cluster member priority for each Virtual Systems in the list.

Note - You cannot use the VSLS configuration file to add or remove cluster members. You must use the appropriate vsx_util commands to accomplish this.

You can use the VSLS configuration file to change member priorities for Virtual Systems after adding or removing a member.

VSLS Configuration File

The VSLS configuration file is a comma separated value (CSV) text file that contains configuration settings for all Virtual Systems controlled by a management server. All lines preceded by the # symbol are comments and are not imported into the management database.

```
# Check Point VSX - VS Load Sharing configuration file
#
# Administrator        : aa
# SmartCenter/Main Domain Management Server : 192.168.50.160
# Generated on         : Thu Jul 23 13:08:42 2009
#
#
#
# VSID, Weight, Active member, Standby member, Backup member #1
# Virtual System name: vs1
2,10,gw150,gw151,gw152

# Virtual System name: vs2
3,10,gw151,gw152,gw150

# Virtual System name: vs3
4,10,gw152,gw150,gw151

# Virtual System name: vs4
6,10,gw151,gw150,gw152

# Virtual System name: vs5
5,10,gw150,gw152,gw151
```
The configuration file contains one line for each Virtual System, consisting of the following data as shown below:

<table>
<thead>
<tr>
<th>VSID</th>
<th>Weight</th>
<th>Primary Member</th>
<th>Standby Member</th>
<th>Backup Member</th>
</tr>
</thead>
<tbody>
<tr>
<td>5,</td>
<td>10,</td>
<td>gw150,</td>
<td>gw152,</td>
<td>gw151</td>
</tr>
</tbody>
</table>

Each line contain the VSID, the weight assigned the Virtual System, one primary member and one standby member. Additional backup members are listed following the standby member.

**Exporting a VSLS configuration**

The most common way to use VSLS configuration files is to initially define your cluster environment and Virtual Systems using SmartDashboard.

To export a VSLS configuration to a text file:
1. From the VSLS menu, select "6. Export configuration to a file".
2. Enter a file name, include its fully qualified path, for example:
   `/home/admin/MyConfiguration`

**Processing Options**

You can insert the following commands in the VSLS Configuration file to display audit trail information while validating and processing data. Each of the commands act as a toggle, whereby the first occurrence of a command enables the action and the next occurrence disables it. These options allow you to efficiently debug very long configuration files by displaying or logging only suspicious sections of the data.

<table>
<thead>
<tr>
<th>Command</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>!comments</td>
<td>Sequentially displays comment lines (those preceded with the ‘#’ character) contained in the configuration file. You can insert comments into the configuration file to indicate which Virtual Systems are currently being processed or to provide status information as the parser processes the data.</td>
</tr>
<tr>
<td>!verbose</td>
<td>Displays whether or not each data line has been successfully verified and the configuration parameters for each Virtual System.</td>
</tr>
<tr>
<td>!log</td>
<td>Saves !comments and !verbose information in the vsx_util.log file.</td>
</tr>
</tbody>
</table>

**Importing a VSLS configuration**

To import a VSLS configuration from a text file:
1. From the VSLS menu, select "5. Import configuration from a file".
2. Enter the file name, include its fully qualified path, for example:
   `/home/admin/MyConfiguration`
3. At the "Save & apply configuration?" prompt, enter "y" to continue.

During the import process, the parser reads the configuration file and attempts to validate the contents. Errors are displayed on the screen together with the offending line number. If either the `!comments` or `!verbose` processing options are enabled, the appropriate information appears on the screen.

The process update process may take several minutes or longer to complete, depending on the quantity of Virtual Systems, Domain Management Servers and cluster members.

Configuring Virtual Systems in Bridge Mode

This section explains configurations and procedures for Virtual Systems in Bridge mode. With native layer-2 bridging instead of IP routing, you can add Virtual Systems without affecting the existing IP structure.

When in Bridge mode, Virtual System interfaces do not require IP addresses. You can assign an IP address to the Virtual System itself (not the interfaces) to enable layer-3 monitoring. This feature enhances network fault detection.

VSX supports these Bridge mode models:

- **STP Bridge Mode**: Redundancy and prevention of undesirable loops between redundant switches.
- **Active/Standby Bridge Mode**: Path redundancy and loop prevention, with seamless support for Virtual System Load Sharing. This model overcomes many of the limitations of STP.

Overview

**STP Bridge Mode**

This section presents the procedures for enabling and configuring the STP Bridge mode for Virtual Systems and VSX Gateways.

The same procedures are applicable for a VSX cluster for PVST + Load Sharing.

**Defining the Spanning Tree Structure**

Define and configure the Spanning Tree structure according to your network requirements. (For PVST + Load Sharing, configure the structure for each VLAN.)

See your hardware documentation for the specific procedures for your network deployment.

**Enabling Active/Active Bridge Mode when Creating Member**

When you create a new VSX Gateway to use as a cluster member, configure it as a cluster member when you first define the gateway.

1. Run: `cpconfig`
2. At `Would you like to install a Check Point clustering product, enter: y`
3. If prompted to disable Active/Standby Bridge Mode, enter: `n`
4. Continue with the cpconfig options as usual.
Enabling Active/Active Bridge Mode for Existing Members

To enable the Active/Active Bridge mode for existing cluster members:

1. Run: cpconfig
2. Enable cluster membership for this member.
   (If a numerical value appears here, cluster membership has already been enabled).
3. Disable Clusters for Bridge Active/Standby.
4. Reboot the member.

Configuring Clusters for Active/Active Bridge Mode

To enable the Active/Active Bridge mode for a cluster:

1. Open SmartDashboard.
2. From the Network Objects tree, double-click the VSX Cluster object.
   The VSX Cluster Properties window opens.
3. Select Other > VSX Bridge Configuration.

Configuring Virtual Systems for STP Bridge Mode

To configure a Virtual System to use bridge mode, define it as a Virtual System in bridge mode when you first create it. You cannot reconfigure a non-Bridge mode Virtual System to use bridge mode later.

Configuring PVST + Load Sharing

Defining the Spanning Tree Structure

Define and configure the Spanning Tree structure for each VLAN according to your network deployment. Please refer to your network hardware documentation for specific procedures.

Configuring a Cluster for PVST + Load Sharing

To configure a VSX cluster for PVST + Load Sharing, perform the procedures described in the STP Bridge Mode section (“STP Bridge Mode” on page 106).

Active/Standby Bridge Mode

This section presents the procedures for enabling and configuring the Active/Standby Bridge Mode for Virtual Systems and VSX Gateways.

Enabling and Configuring Active/Standby Bridge Mode

Enabling Active/Standby Bridge Mode for a New Member

When you create a new cluster member, enable the cluster options during the first configuration.

1. In the Gaia First Time Configuration Wizard Products page, select ClusterXL.
2. From the VSX Gateway CLI, run: cpconfig
   - If you enable the Per Virtual System State feature, (required for VSLS), Active/Standby Bridge Mode is enabled automatically.
• If you chose not to enable Virtual System Load Sharing, an option to enable **Active/Standby Bridge Mode** appears. Enter `y` and continue with the gateway configuration.

*Enabling Active/Standby Bridge Mode for Existing Members*

To enable the Active/Standby Bridge Mode on existing Virtual Systems:

1. Run: `cpconfig`
2. Enable **ClusterXL for Bridge Active/Standby**.
3. Reboot the member.

*Configuring Clusters for Active/Standby Bridge Mode*

To enable the Active/Standby Bridge Mode for a cluster:

1. Open SmartDashboard.
2. From the Network Objects tree, double-click the VSX Cluster object. The **VSX Cluster Properties** window opens.
3. Select **Other > VSX Bridge Configuration**.
4. Select **Check Point ClusterXL**. The Active/Standby Bridge Mode loop detection algorithms in ClusterXL is enabled.

*Configuring Virtual Systems for Active/Standby Bridge Mode*

To configure a Virtual System to use bridge mode, define it as such when you first create the object.

To configure a Virtual System for the Active/Standby Bridge Mode:

1. In the **Virtual System General Properties** page of the new Virtual System object, select **Bridge Mode**.
2. Click **Next**. The **Virtual System Network Configuration** window opens.
3. Configure the external and internal interfaces for the Virtual System.
4. **Optional**: Select **Enable Layer-3 Bridge Interface Monitoring**.
   - The IP address must be unique and on the same subnet as the protected network.
5. Click **Next** and then click **Finish**.

*Advanced Clustering Configuration*

This section presents several advanced cluster scenarios and procedures for their configuration.

*Clusters on the Same Layer-2 Segment*

The recommended cluster architecture contains interfaces connect to a Layer-2 segment that is isolated from other clusters. When configuring a cluster with only two members, you should connect the secured interfaces of the sync network using a crossover cable.

However, in a deployment where multiple clusters need to connect to the same Layer-2 segment, the same MAC address may be used by more than one cluster for Cluster Control Protocol (CCP) communication. This may direct traffic to the incorrect cluster. In this case you will need to modify the source MAC address(es) of the clusters.
This section describes how source MAC addresses are assigned, and explains how to change them. This procedure applies to both ClusterXL and OPSEC certified clustering products using the High Availability mode.

**Source Cluster MAC Addresses**

Cluster members use CCP to communicate with each other. In order to distinguish CCP packets from ordinary network traffic, CCP packets are given a unique source MAC address.

- The first four bytes of the source MAC address are all zero: 00.00.00.00
- The fifth byte of the source MAC address is a “magic” number, a number that encodes critical information in a way intended to be opaque. Its value indicates its purpose:

<table>
<thead>
<tr>
<th>Default Value Of Fifth Byte</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xfe</td>
<td>CCP traffic</td>
</tr>
<tr>
<td>0xfd</td>
<td>Forwarding layer traffic</td>
</tr>
</tbody>
</table>

- The sixth byte is the ID of the source cluster member

When multiple clusters are connected to the same Layer-2 segment, setting a unique value to the fifth byte of the MAC source address of each cluster allows them to coexist on the same Layer-2 segment.

**Changing a Cluster’s MAC Source Address**

To change a cluster’s MAC source address, run these commands on each cluster member:

```shell
fw ctl set int fwha_mac_magic <value>
fw ctl set int fwha_mac_forward_magic <value>
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>fwha_mac_magic</td>
<td>0xfe</td>
</tr>
<tr>
<td>fwha_mac_forward_magic</td>
<td>0xfd</td>
</tr>
</tbody>
</table>

Use any value, as long as the two gateway configuration parameters are different. To avoid confusion, do not use the value 0x00.

**Making the Change Permanent**

You can configure the above parameters to persist following reboot.

1. Use a text editor to open the file `fwkern.conf`, located at `$FWDIR/boot/modules/`.
2. Add the line `Parameter=<value in hex>`. Make sure there are no spaces.

**Monitoring all VLANs with ClusterXL**

By default, ClusterXL only monitors two VLANS for failure detection and failover. These are the highest and lowest VLAN tags defined for a given interface.

For example, if the topology for interface eth1 includes several VLAN tags in the range of eth1.10 to eth1.50, ClusterXL only monitors VLANS eth1.10 and eth1.50 for failure. Failures on any of the other VLANs are not detected in the default configuration.

Note - The command line option `cphaprob -a if` displays the highest and lowest VLANs being monitored.
When both the highest and lowest VLANs fail, all the VLANs are considered down, and a failover occurs. This means that if a VLAN which is not listed as the highest or lowest goes down, the trunk is still considered "up", and no failover occurs.

There are instances in which it would be advantageous to monitor all the VLANs in the trunk, not just the highest and lowest, and initiate a failover when any one of the VLANs goes down.

To enable monitoring of all VLANs, enable the `fwha_monitor_all_vlan` property in $FWDIR/boot/modules/fwkern.conf. Change the property to `fwha_monitor_all_vlan=1`.

Note - Monitoring all VLANs is enabled automatically when the Per VLAN state option is enabled.

Enabling Broadcast Mode

The default ClusterXL Control Protocol transport mode is multicast. Use the `cphaconf set_ccp` command to configure broadcast or multicast mode for the cluster.

To enable broadcast or multicast mode:

1. On a cluster member, run `cphaconf set_ccp {broadcast|multicast}`
2. Do the previous steps for all the cluster members.
Deploying VSX

In This Section:

- Introduction ................................................................. 111
- Internal Network Deployment Strategies ......................... 111
- Organizational Deployment Strategies .............................. 117
- Migrating between Servers with Different Interface Names ...... 123

This chapter presents deployment concepts and strategies for exploiting VSX virtualization and its unique feature set. This presentation is divided into two sections as follows:

- **Internal Network Protection Strategies:** Presents concepts and examples for protecting internal network resources, including a comparison between physical deployments and VSX virtualization.

- **Organizational Deployment Strategies:** Presents VSX deployment strategies and features for several different types of large organizations.

Introduction

This chapter presents deployment concepts and strategies for exploiting VSX virtualization and its unique feature set. This presentation is divided into two sections as follows:

- **Internal Network Protection Strategies:** Presents concepts and examples for protecting internal network resources, including a comparison between physical deployments and VSX virtualization.

- **Organizational Deployment Strategies:** Presents VSX deployment strategies and features for several different types of large organizations.

Internal Network Deployment Strategies

Security Gateway Deployment on a Physical Network

In large physical network deployments, multiple Check Point security products, such as Security Gateways or UTM-1 Edge appliances, are deployed to protect various network segments.
Each Security Gateway physically connects to its own internal protected network as well as to a router for access to other internal networks and the Internet.

**VSX Virtual System Deployment Strategies**

In a VSX environment, Virtual Systems protect internal networks, much in the same manner as Security Gateways and other Check Point products in a physical network. This section presents several sample VSX deployments using Virtual Systems to protect internal networks.

Each example highlights certain VSX features. In a real-world deployment, you can combine several of the concepts presented in these examples to create a powerful network security solution for complex enterprise environments.

**Physical Internal Interface for Each Virtual System**

The figure below shows a basic VSX configuration where Virtual Systems connect directly to protected internal networks using physical interfaces on the VSX Gateway. A Virtual Switch provides connectivity between internal networks, as well as to the Internet. This deployment is simple to provision and is suitable for protecting a small, fixed quantity of internal networks.

The main disadvantage of this deployment is that each protected network requires its own dedicated physical interface on the VSX Gateway. Obviously, this deployment is not suitable for networks that require many Virtual Systems.

**Virtual Systems with Internal VLAN Interfaces**

In this deployment example, Virtual Systems connect to internal protected networks using VLAN interfaces. The VSX Gateway connects to a VLAN switch via an 802.1q VLAN trunk, which is an aggregate of all VLANs passing through it.
This deployment option is appropriate for environments where many Virtual Systems protect many internal networks with a single VSX Gateway or cluster. The use of VLANs provides scalability as well as granularity, allowing administrators to provision additional Virtual Systems and protected networks quickly and without impacting the existing IP address structure.

### Internal Virtual Router with Source-Based Routing

This deployment scenario enables Virtual Systems to connect to protected networks using a single physical interface without VLAN technology. The Virtual Router uses source-based routing rules to forward traffic to the appropriate Virtual System based on its source IP address.

In a VSX deployment with each Virtual System connected to a single Virtual Router: You can configure the Virtual Router to use source-based routing rules, to forward traffic to the appropriate Virtual System, based on the source IP address.
Notes to this scenario:

- Each Virtual System uses a public IP address to connect to the Virtual Switch.
- Each local network connected to a Virtual Router uses private IP addresses.
- This deployment does not support overlapping IP addresses.
- Anti-Spoofing protection does function for packets originating from the shared internal interface. We recommend that you configure the internal physical router to perform Anti-Spoofing protection.

The Routing Concept section ("VSX Routing Concepts" on page 27) provides a detailed discussion of routing options in VSX environments.

**Virtual Systems in the Bridge Mode**

A Virtual System in the bridge mode implements native layer-2 bridging instead of IP routing. This allows network administrators to easily and transparently deploy a Virtual System in an existing network topology without reconfiguring the existing IP routing scheme. The figure below, shows a scenario where each Virtual System in the Bridge Mode protects a VLAN switched network.

Bridge Mode (on page 81) deployments are particularly suitable for large-scale clustered environments.

**Cluster Deployments**

This section presents several examples of cluster deployments that highlight important VSX features. The discussion is intended to introduce these features as they relate to deployment strategy. Refer to the conceptual discussion of cluster deployments ("Introduction to VSX Clusters" on page 74) section for more information.
Active/Standby Bridge Mode

The Active/Standby Bridge Mode provides path redundancy and loop prevention, while offering seamless support for Virtual System Load Sharing and overcoming many Spanning Tree Protocol (STP) Bridge mode limitations.

VLAN Shared Interface Deployment - Active/Standby Bridge Mode

In this scenario, each individual member connects to a pair of redundant switches via a VLAN trunk. All Virtual Systems in a given member share the same VLAN trunk.

When using the Active/Standby Bridge Mode in a High Availability deployment, VSX directs traffic to members according to predefined priorities and member status. In VSLS deployments, VSX distributes the traffic load amongst members according to a set of predefined preferences.

This deployment scenario is appropriate for very large enterprises.

Virtual Systems in Bridge Mode

A three layer hierarchical model is appropriate for large, high-traffic network environments. It contains a mixture of components as described below:

1. A core network, comprised of high-speed backbone switches directs traffic to and from the Internet and other external networks.
2. A distribution layer, comprised of routers, provides connectivity between the core and the access layer.
3. An access layer, comprised of redundant LAN switches, forwards traffic to and from internal networks.
Use Active/Standby Bridge Mode with VSX to enforce the security policy over the distribution layer.

The routers direct external, "dirty" traffic (typically from the Internet) to the appropriate Virtual System via a segregated VLAN. Filtered, "clean" traffic exits the Virtual System through a separate segregated VLAN back to the routers and on to internal destinations.

This deployment scenario is appropriate for very large enterprises.

**Virtual System Load Sharing (VSLS)**

VSX clusters can efficiently balance network traffic load by distributing active Virtual Systems amongst cluster members. This capability is known as **Virtual System Load Sharing (VSLS)**.

In a deployment scenario with three cluster members, each with three Virtual Systems: an equalized Load Sharing deployment might have one active Virtual System on each cluster member.

A different member hosts the active peer for each Virtual System. This distribution spreads the load equally amongst the members. When you create a Virtual System, VSX automatically assigns
standby and backup states to the appropriate peers and distributes them among the other cluster members.

In the event that a cluster member fails, VSLS directs traffic destined to affected Virtual Systems to their fully synchronized standby peers, which then become active. At the same time, a backup Virtual Systems switches to standby, and synchronizes with the newly active Virtual System.

In the event that an individual active Virtual System fails, it immediately fails over to its standby peer and one of its backup peers becomes the standby, synchronizing with the newly active peer.

Organizational Deployment Strategies

This section presents deployment scenarios for different types of large organizations and illustrates how VSX provides security both internally and at the perimeter. The discussion covers the following types of organizations:

- Large Enterprises
- Managed Service Providers
- Data Centers

Enterprise Deployments

Large enterprise network environments typically contain a wide variety of diverse networks, distributed over multiple locations around the world. These networks often have differing security and access requirements for various departments and branches. The ability to centrally manage network security while maintaining throughput is a critical requirement.
Core Network Security

Many Enterprise environments are based on core networks. Situated adjacent to core network backbone switches, VSX protects the internal network by providing security at layer-2, layer-3 or both. VSX communicates with the core network using the existing infrastructure. With Virtual Systems in the Bridge Mode, VSX can protect departmental networks, while simultaneously preventing network segmentation. In this case, switches are located at the entrance to each department’s network.

VSX ensures connectivity between the core network and the Internet or external networks, while providing perimeter security. Security can be configured on a per VLAN basis.

Dynamic Routing

In an enterprise network with dynamic routing protocols (OSPF/BGP), VSX secures the DMZ services, VPN peers, Domains and partner networks.
In this example, BGP neighbor updates in the routed core network are selectively redistributed to application networks. OSPF provides connectivity between Virtual Routers, Virtual Systems, the core network and application networks.
**Perimeter Security**

For example, security is enforced on each VLAN. The OSPF and BGP Dynamic routing protocols provide connectivity to multiple security zones along the perimeter.

**Notes to this scenario:**

- Partners access network resources remotely via Virtual Systems
- Each Virtual System has its own security policy based on its requirements
- Logs and audit information for each partner is collected separately, and saved to a private database
- Applications and services are segregated by private Virtual Systems
- Multiple Virtual Routers / Virtual Switches are used to control the access paths

**Managed Service Providers Using Multi-Domain Security Management**

Managed service providers give connectivity and security services for Domain networks. Some of these Domains require remote access capabilities. In this service oriented environment, VSX and Multi-Domain Security Management provide central management and make connectivity and security easier, without affecting the existing IP topology.
In this scenario, a VSX cluster is in a Point of Presence (POP) deployment for a service provider. VSX consolidates hardware for the service provider and ensures privacy and secure connectivity solutions (VPN) for users. This scenario is appropriate for High Availability and Virtual System Load Sharing cluster modes.

VSX and Multi-Domain Security Management provide a centralized, granular provisioning system for a number of Domains. Applications and services are separated by discrete Virtual Systems. Access to these services and applications is based on need.

**Scenario:**

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Internet. Routers are between the VSX cluster members and the Internet.</td>
</tr>
<tr>
<td>2</td>
<td>VSX cluster. One member handles the Local Exchange and another handles server traffic of different Domains.</td>
</tr>
<tr>
<td>3</td>
<td>Core IP VPN Network.</td>
</tr>
<tr>
<td>4</td>
<td>Multi-Domain Security Management at the Network Operation Center monitors POP and connects to VSX Gateway. The Multi-Domain Log Server in the NOC collects data for each Domain and stores the logs in separate private databases.</td>
</tr>
<tr>
<td>5</td>
<td>Multi-Domain Security Management at the NOC and the VSX Gateway make the Local Exchange.</td>
</tr>
<tr>
<td>Item</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>6</td>
<td>Domain A web servers.</td>
</tr>
<tr>
<td>7</td>
<td>Domain B DMZ.</td>
</tr>
<tr>
<td>8</td>
<td>Domain C mail servers.</td>
</tr>
<tr>
<td>9</td>
<td>PE Router.</td>
</tr>
<tr>
<td>10, 11, 12</td>
<td>Domain A, B, and C. Each Domain manages its own security and cannot define Virtual Systems or other network components. Domains have secure VPN connectivity.</td>
</tr>
</tbody>
</table>

**Data Centers**

Data center providers supply external hosting services for Domain servers and databases. The service typically includes infrastructure, connectivity, and security for multiple Domains.

For example, you can have a scenario such as:

- Multiple Domain networks sharing a common physical infrastructure.
- Backbone that provides connectivity between each Domain and the data center.
- Domain A connects to its web hosting servers.
- Domain B connects to its mail servers.
- Domain C connects to its database servers.

To provide network security and management, the data center provider deploys a VSX Gateway with one Virtual System for each Domain.

This scenario offers a cost effective scalability solution for network expansion by means of remote connectivity. In this example, a VPN connection between a Domain Virtual System and a UTM-1 Edge device protecting a remote network, integrates that network into the MPLS core. Similarly, a Virtual System can provide access for individual remote users who connect intermittently.

**Data Centers in an Enterprise**

This example scenario illustrates how VSX provides security management for enterprise data centers. By assigning layer-2 connections to Virtual Systems, VSX reduces the number of physically managed devices within a data center while providing the same high level of security.
For example, a VSX Gateway allows authorized users to access data center resources. The objective here is to protect shared resources with differing access permissions and security requirements, while implementing network granularity.

For example, one Virtual System protects databases against SQL vulnerabilities. Another Virtual System protects Web Servers using IPS. When new applications and services are added to the enterprise data center, new Virtual Systems are easily created to secure them according to their specific requirements.

Migrating between Servers with Different Interface Names

Check Point VSX-1 appliances use different interface names than Open Server platforms (Gaia, Linux). When migrating an Open Server VSX Gateway or cluster to a VSX-1 appliance, you must use the `vsx_util change_interfaces` command ("change_interfaces" on page 175) to change the appliance interface names.

The `vsx_util change_interfaces` command ("change_interfaces" on page 175) command contains a **Management Only** option that allows you to change the interface names on the management server (Security Management Server or Multi-Domain Security Management Domain Management Server) only. You then use the `vsx_util reconfigure` command ("reconfigure" on page 179) to push the updated configuration to VSX Gateways or cluster members.

To migrate a VSX Gateway or cluster to a VSX-1 appliance:


2. On the management server, enter the **Expert Mode** and run the `vsx_util change_interfaces` command.
3. When prompted, enter the Security Management Server, or Multi-Domain Security Management Main Domain Management Server IP address.
4. When prompted, enter the administrator name and password as prompted.
5. When prompted, enter the VSX cluster object name.
6. When prompted, select the **Management Only** option.
7. When prompted, select the interface to be replaced.
8. When prompted, enter 1 and then enter the new interface name.
9. When prompted to change another interface, enter "y" when prompted and repeat steps 7 and 8 as required.
10. To complete the process, enter "n".
11. When prompted to remove the old interfaces from the database enter "y".
12. On the VSX Gateway or cluster members, run the `vsx_util reconfigure` ["reconfigure" on page 179] command.
13. Reboot the VSX Gateway or cluster members.
Working with Link Aggregation

In This Section:
- Link Aggregation Overview ................................................................. 125
- Configuring High Availability Mode ...................................................... 129
- Configuring Load Sharing Mode ............................................................. 131
- Configuring Cisco Switches for Load Sharing ....................................... 133
- Troubleshooting Bonded Interfaces ....................................................... 134
- Gaia CLI Reference ................................................................................ 135

Link aggregation, also known as interface bonding, joins multiple physical interfaces to be one virtual interface: a **bond interface**. Configure a bond interface for High Availability (redundancy), or for load sharing (increased throughput).

For more about Link Aggregation, see the R76 ClusterXL Administration Guide.

Link Aggregation Overview

Link aggregation, also known as interface bonding, joins multiple physical interfaces together into a virtual interface, known as a **bond interface**. A bond interface can be configured for High Availability redundancy or for load sharing, which increases connection throughput above that which is possible using one physical interface.

For more about Link Aggregation, see the R76 ClusterXL Administration Guide http://supportcontent.checkpoint.com/documentation_download?ID=22910.

Link Aggregation Terminology

- **Link Aggregation (Interface Bonding)**: Networking technology that binds multiple physical interfaces together into one virtual interface.

- **Bond**: A group of physical interfaces that operate together as one virtual interface and share an IP address and MAC address. A bond is identified by the cluster by its **Bond ID** (for example: bond0).

- **Bond Interface**: The logical representation of the bond.

- **Slave (enslaved interface)**: A physical interface that is a member of a bond. Slaves do not have an IP Address and in some cases share the same MAC address.
How Link Aggregation Works

A bond contains a minimum of one and may contain up to eight slave interfaces. All slave interfaces contained in a bond share a common IP address and may share the same MAC address. We recommend that each cluster member contain the same quantity of identical slave interfaces.

You can configure Link Aggregation using one of the following strategies:

- **High Availability (Active/Backup)**: Ensures redundancy in the event of interface or link failure. This option also provides switch redundancy.

- **Load Sharing (Active/Active)**: All interfaces are active, but handle different connections simultaneously. Traffic is balanced amongst slave interfaces to maximize throughput. The Load Sharing option does not support switch redundancy.

High Availability Overview

Clusters, by definition, provide redundancy and high availability at the gateway level. Link Aggregation, however, adds interface and switch redundancy by providing automatic failover to a standby interface card within the same VSX Gateway.

In a High Availability deployment, only one interface is active at a time. If an interface or connection fails, the bond fails over to a standby slave interface. Bonding High Availability failover occurs in one of these cases:

- An active interface detects a link state failure in a monitored interface.
- ClusterXL detects a failure in sending or receiving Cluster Control Protocol (CCP) keep-alive packets.
**Fully Meshed Redundancy via Interface Bonding**

The Link Aggregation High Availability mode, when deployed with ClusterXL, enables a higher level of reliability by providing granular redundancy in the network. This granular redundancy is achieved by using a fully meshed topology, which provides for independent backups for both NICs and switches.

![Diagram of Fully Meshed Redundancy](image)

In this scenario:
- Member-1 and Member-2 are cluster members in the High Availability mode
- S-1 and S-2 are switches
- C-1, C-2, C-3 and C-4 are network connections

**Load Sharing Overview**

Load sharing provides the ability to spread traffic over multiple slave interfaces, in addition to providing interface redundancy. All interfaces are always active.

Traffic is balanced between interfaces in a manner similar to the way load sharing balances traffic between cluster members. Load sharing operates according to either the IEEE 802.3ad or the XOR standard.

In Load Sharing mode, each individual connection is assigned to a specific slave interface. For a specific connection, only the designated slave interface is active. In the event of a failure of the designated slave interface, the traffic fails over to another interface, adding that connection’s to the traffic it is already handling.

**Bond Failover**

Either of the following failure scenarios can induce bond failover:
- An active interface detects a link state failure in another monitored interface
- ClusterXL detects a failure in sending or receiving Cluster Control Protocol (CCP) keep-alive packets

Either of these occurrences will induce a failover, either to another slave interface within the bond, or between cluster members, depending on the circumstances.
Note - The bond failover operation requires a network interface card that supports the Media-Independent Interface (MII) standard.

Link State Initiated Failover

Link-state initiated failover occurs in this sequence:

1. The active slave interface detects a **down** link state.
2. The bond initiates failover to a standby interface. Since this is a failover within the bond, the status of the other cluster member is unaffected. When the number of available slave interfaces is fewer than the critical minimum number of interfaces ("Setting Critical Required Interfaces" on page 132), failover to other cluster members occurs.
3. If the standby interface continues to detect a link failure, and the initial interface is still down, failover to other cluster members occurs.

CCP Initiated Failover

CCP failover occurs only when other cluster members are not down, in this sequence.

1. ClusterXL detects a problem sending or receiving of CCP packets.
2. ClusterXL initiates an internal bond failover.
3. ClusterXL monitors CCP packet transmission and reception. If additional problems are detected within three minutes, the system initiates a failover to another cluster member.

Failover Support for VLANs

ClusterXL monitors VLAN IDs for connectivity failure or miscommunication, and initiates failover when necessary. By default, both the highest and the lowest VLAN IDs are monitored for failure. This is done by sending ClusterXL Control Protocol (CCP) packets on round-trip paths at a set interval.

You can configure VSX to monitor all VLANs.
When a failure is detected, a log of the failure is recorded in SmartView Tracker.

**Monitoring the Highest and Lowest VLAN IDs**

By default, the highest and lowest VLAN IDs indicate the status of the physical connection. These VLAN IDs are always monitored and a connectivity failure in either initiates a failover. In most deployments this is the desired setting, as it supports the primary purpose of the feature (detecting a connectivity failure) and the traffic generated on the network is light. However, this setting only detects VLAN configuration problems on the switch for the highest and lowest VLAN IDs.

**Bond Interface & Interface Limitations**

- You can define a maximum of 4096 interfaces on a Gaia server or appliance. The total number of bond interfaces in use is the sum of bonds plus the number of slave interfaces contained in each bond.
- Up to eight interfaces can be defined in a Link Aggregation deployment for each bond interface.

**Configuring High Availability Mode**

This section explains how to configure High Availability on a bond interface. Run the CLI commands from the VSX Gateway (VS0) context. For a cluster configuration, run these commands on each cluster member.

Use the `active-backup` value for the `mode` parameter to configure High Availability.

**Configuring the High Availability Bond**

This is a workflow of CLI commands to configure Link Aggregation in High Availability mode. When you are enslaving configured interfaces, make sure that these interfaces are not used in other configurations.

To configure High Availability:

1. Create the High Availability bond. Run:
   ```
   add bonding group <bond id>
   set bonding group <bond id> mode active-backup
   ```
2. Define the slave interfaces. Run:
   ```
   add bonding group <bond id> interface <IF name>
   ```
   Do this command again for all of the slave interfaces.
3. Make sure that the bond is configured correctly. Run:
   ```
   show bonding group <bond id>
   ```
   To show more information about the bond, from Expert mode run:
   ```
   cat /proc/net/bonding/<bond id>
   ```
4. Open SmartDashboard and configure the cluster object.
   - For a new Link Aggregation installation, create a new cluster object.
   - For updating an existing configuration, update the interface topology ("Updating the Interface Topology" on page 130).
**Updating the Interface Topology**

When you are updating an existing configuration to Link Aggregation, it is necessary to reconfigure the relevant objects to connect to the newly created bond. This includes Virtual Systems, Virtual Routers and Virtual Switches. You can perform these actions using SmartDashboard. In most cases, these definitions can be found in the object Properties window.

For large existing VSX deployments containing many Domain Management Servers and virtual devices, use the `vsx_util change_interfaces` command to reconfigure existing object topologies. For example, in a Multi-Domain Security Management deployment with 200 Domains, each with many virtual devices, it is faster to use `vsx_util change_interfaces`. This command automatically replaces the interface with the new bond on all relevant objects.

**Reconfiguring the Bond using SmartDashboard**

To configure the newly created bond:

1. Open SmartDashboard.
2. From the **Network Objects** tree, double-click the VSX Gateway or cluster object.
3. From the navigation tree, click **Physical Interfaces**.
4. Click **Add**.
   - The **Physical Interface Properties** window opens.
     a) Enter the bond name.
     a) If the bond is a VLAN trunk, select **VLAN Trunk**.
     b) Click **OK**.
5. From the navigation tree, click **Topology**.
6. Do these steps for each interface that you are adding to the bond.
   a) Double-click the interface.
   - The **Interface Properties** window opens.
   b) From **Interface**, select the bond interface.
   c) Click **OK**.
7. Install the policy.
8. Delete the slave interfaces of the newly created bond that you are not using.
   You can also replace a bond interface with one that is being used ("Reconfiguring Topology with vsx_util change_interfaces" on page 130).

**Reconfiguring Topology with vsx_util change_interfaces**

To reconfigure objects with `vsx_util change_interfaces`:

⚠️ **Important** - In a Multi-Domain Security Management environment, all Domain Management Servers must be unlocked in order for this operation to succeed.

2. On the management server, enter the Expert Mode and execute the `vsx_util change_interfaces` command.
3. Enter the Security Management Server, or Multi-Domain Security Management main Domain Management Server IP address.
4. Enter the administrator name and password as requested.
5. Enter the VSX cluster object name.
6. Select Apply changes to the management database and to the VSX Gateway/Cluster members immediately.
7. When prompted, select the interface to be replaced.
8. When prompted, select the replacement bond interface.
9. If you wish to replace additional interfaces, enter "y" when prompted and repeat steps 6 and 7.
10. To complete the process, enter "n".

Configuring Load Sharing Mode

This section explains how to configure Load Sharing on a bond interface. Run the CLI commands from the VSX Gateway (VS0) context. For a cluster configuration, run these commands on each cluster member.

Configure one of these Load Sharing modes for the bond interface:
- Round robin
- XOR
- 802.3ad

Configuring the Load Sharing Bond

This is a workflow of CLI commands to configure Link Aggregation in Load Sharing mode.

When you are enslaving configured interfaces, make sure that these interfaces are not used in other configurations.

To configure Load Sharing:

1. Create the High Availability bond. Run:
   ```
   add bonding group <bond id>
   set bonding group <bond id> mode <round-robin|xor|8023AD>
   ```
2. Define the slave interfaces. Run ```add bonding group <bond id> interface <IF name>``` Do this command again for all of the slave interfaces.
3. Define the number of critical interfaces ([“Setting Critical Required Interfaces” on page 132]).
4. For configurations that use Performance Pack, configure the core affinities ([“Setting Affinities” on page 132]).
5. Make sure that the bond is configured correctly. Run ```show bonding group <bond id>``` To show more information about the bond, from Expert mode run ```cat /proc/net/bonding/<bond id>```
6. Open SmartDashboard and configure the cluster object.
   - For a new Link Aggregation installation, create a new cluster object.
   - For updating an existing configuration, update the interface topology ([“Updating the Interface Topology” on page 130]).
Setting Critical Required Interfaces

**Note** - The Critical Required Interfaces feature is supported for ClusterXL only.

A bond in Load Sharing mode is considered to be down when fewer than a critical minimum number of slave interfaces remain up. When not explicitly defined, the critical minimum number of interfaces in a bond of n interfaces is n-1. Failure of a second interface will cause the entire bond to be considered down, even if the bond contains more than two interfaces.

If a smaller number of interfaces will be able to handle the expected traffic, you can increase redundancy by explicitly defining the number of critical interfaces. Divide your maximal expected traffic speed by the speed of your interfaces and round up to a whole number to determine an appropriate number of critical interfaces.

To explicitly define the number of critical interfaces, create and edit the following file:

```bash
$FWDIR/conf/cpha_bond_ls_config.conf
```

Each line of the file should be of the following syntax:

```bash
<bondname> <critical#>
```

For example, if bond0 has seven interfaces and bond1 has six interfaces, file contents could be:

- bond0 5
- bond1 3

In this case bond0 would be considered down when three of its interfaces have failed. bond1 would be considered down when four of its interfaces have failed.

Setting Affinities

If you are running Performance Pack in a multi-core system, after you define bonds, set affinities manually. Use the `-s` parameter of the `sim affinity` command, see the *R76 Performance Pack Administration Guide*.

**Note** - `sim affinity` commands take effect only if the Performance Pack is enabled and actually running. Performance Pack begins running when you install a policy for the first time.

For optimal performance, set affinities according to the following guidelines:

1. Run `sim affinity` using the `-s` option.
2. Whenever possible, dedicate one processing core to each interface. See sk33520 http://supportcontent.checkpoint.com/solutions?id=sk33250.
3. If there are more interfaces than cores, one or more cores handle two interfaces. Use interface pairs of the same position with internal and external bonds.
   a) To view interface positions in a bond, run:
   ```bash
cat /proc/net/bonding/<bond name>.
```
   b) Note the sequence of the interfaces in the output, and compare this for the two bonds (external bond and its respective internal bond). Interfaces that appear in the same position in the two bonds are interface pairs and set to be handled by one processing core.
For example, you might have four processing cores (0-3) and six interfaces (0-5), distributed among two bonds:

<table>
<thead>
<tr>
<th>bond0</th>
<th>bond1</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth0</td>
<td>eth3</td>
</tr>
<tr>
<td>eth1</td>
<td>eth4</td>
</tr>
<tr>
<td>eth2</td>
<td>eth5</td>
</tr>
</tbody>
</table>

Two of the cores will need to handle two interfaces each. An optimal configuration might be:

<table>
<thead>
<tr>
<th>bond0</th>
<th>bond1</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth0</td>
<td>core 0</td>
</tr>
<tr>
<td>eth1</td>
<td>core 1</td>
</tr>
<tr>
<td>eth2</td>
<td>core 2</td>
</tr>
<tr>
<td></td>
<td>eth5</td>
</tr>
<tr>
<td></td>
<td>core 3</td>
</tr>
</tbody>
</table>

### Configuring Cisco Switches for Load Sharing

These are sample configuration commands for Cisco switches.

**For 802.3ad:**

```bash
Switch#conf t
Switch(config)#port-channel load-balance src-dst-ip
Switch(config)#interface FastEthernet <all the participating interfaces>
Switch(config-if)#channel-group 1 mode active
Switch(config-if)#channel-protocol lacp
Switch(config-if)#exit
Switch(config)#interface port-channel 1
Switch(config-if)#switchport access vlan <the wanted vlan number>
Switch(config-if)#end
Switch#write
```

**For XOR:**

```bash
Switch#conf t
Switch(config)#port-channel load-balance src-dst-ip
Switch(config)#interface FastEthernet <all the participating interfaces>
Switch(config-if)#channel-group 1 mode on
Switch(config-if)#exit
Switch(config)#interface port-channel 1
Switch(config-if)#switchport access vlan <the wanted vlan number>
Switch(config-if)#end
Switch#write
```
Troubleshooting Bonded Interfaces

1. Check the status of the bond. From Expert mode, run `cat/proc/net/bonding/<bond id>`
2. If there is a problem, check if the physical link is down, as follows:
   a) Execute the following command:
      
      ```
      cphaconf show_bond <bond-name>
      ```
   b) Look for a slave interface that reports the status of the link as `no`.
   c) Check the cable connections and other hardware.
   d) Check the port configuration on the switch.

3. Check if a cluster member is down, by running:
   `cphaprob state`
   
   If any of the cluster members have a Firewall State other than active, see Monitoring Cluster Status [cphaprob state] in the R76 ClusterXL Administration Guide http://supportcontent.checkpoint.com/documentation_download?ID=22910.

4. For further information regarding bond status and failovers, view logs in SmartView Tracker.
   Any interface bond status change is logged and can be viewed in SmartView Tracker.

Connectivity Delays on Switches

When using certain switches, connectivity delays may occur during some internal bond failovers. With the various features that are now included on some switches, it can take close to a minute for a switch to begin servicing a newly connected interface. The following are suggestions for reducing the startup time after link failure.

1. Disable auto-negotiation on the relevant interface.
2. On some Cisco switches, enable PortFast, as detailed below.

   ```
   Note - PortFast is not applicable if the bond group on the switch is configured as Trunk.
   ```

Warning Regarding Use of PortFast

The PortFast feature should never be used on ports that connect to other switches or hubs. It is important that the Spanning Tree complete the initialization procedure in these situations. Otherwise, these connections may cause physical loops where packets are continuously forwarded (or even multiply) in such a way that network will ultimately fail.

Sample Configuration of PortFast on a Cisco Switch

The following are the commands necessary to enable PortFast on a Gigabit Ethernet 1/0/15 interface of a Cisco 3750 switch running IOS.

1. Enter configuration mode:
   ```
   cisco-3750A#conf t
   ```
2. Specify the interface to configure:
   ```
   cisco-3750A(config)#interface gigabitethernet1/0/15
   ```
3. Set PortFast on this interface:
   
cisco-3750A(config-if)#spanning-tree portfast

Gaia CLI Reference

This section contains a summary of the Gaia CLI commands that configure Link Aggregation.

Link Aggregation - CLI (bonding)

This section is a quick reference for Link Aggregation commands. The next sections include procedures for different tasks, including explanations of the configuration options.

Use these commands to configure link aggregation.

Syntax:
{add | delete} bonding group <bondID> interface <IFName>

set bonding [group <bondID>] [primary <IFName>] [mii-interval <ms>] [up-delay <ms> | down-delay <ms>] [mode {round-robin | active-backup | xor [xmit-hash-policy {layer2 | layer3+4}] | 8023AD [lacp-rate {slow | fast}]}

show bonding group {<bondID> | groups}

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bondID</td>
<td>ID of bond, an integer between 1 and 1024</td>
</tr>
<tr>
<td>IFName</td>
<td>Name of interface to add to the bond</td>
</tr>
<tr>
<td>primary</td>
<td>Name of primary interface in the bond</td>
</tr>
<tr>
<td>mii-interval</td>
<td>Frequency that the system polls the Media Independent Interface (MII) to get status</td>
</tr>
<tr>
<td>up-delay</td>
<td>Waiting time to confirm the interface status before taking the specified action (0-5000 ms, default = 200 ms)</td>
</tr>
<tr>
<td>down-delay</td>
<td></td>
</tr>
<tr>
<td>mode</td>
<td>Bond operating mode [“Defining the Bond Operating Mode” on page 136]</td>
</tr>
<tr>
<td>lacp-rate</td>
<td>Link Aggregation Control Protocol packet transmission rate:</td>
</tr>
<tr>
<td></td>
<td>• slow - LACPDU packet sent every 30 seconds</td>
</tr>
<tr>
<td></td>
<td>• fast - LACPDU packet sent every second</td>
</tr>
<tr>
<td>xmit-hash-policy</td>
<td>Algorithm for interface selected by TCP/IP layer</td>
</tr>
</tbody>
</table>

Example
set bonding group 666 20 eth2
show bonding groups
Output

Bonding Interface: 20

Bond Configuration:
  xmit_hash_policy Not configured
  down-delay 200
  primary Not configured
  mode round-robin
  up-delay 200
  mii-interval 100
  lacp_rate Not configured

Bond Interfaces:
  eth2
  eth3

Creating or Deleting a Bond Interface

To add a new bond interface:

add bonding group <bondID>

Example:

add bonding group 777

To delete a bond interface:

1. Remove all interfaces from the bond.
2. Run: delete bonding group <bondID>

Defining the Bond Operating Mode

Define how interfaces are activated in a bond:

- `round-robin` - Interfaces activated in order by ID (default)
- `active-backup` - On active interface down, failover to primary interface first, and to other interfaces if primary is down
- `xor` - Interface activation by TCP/IP layer (layer2 or layer3+4).
  You can set the LACP packet transmission rate for xor mode or 8023AD mode. After you set one of these Load Sharing modes, enter this option: `lacp-rate {slow | fast}` where slow is every 30 seconds, and fast is every one second.

- `8023AD` - Link Aggregation Control Protocol load shares traffic by dynamic interface activation, with full interface monitoring between gateway and switch. In this mode only, you can set the algorithm for interface selection, according to the specified TCP/IP layer:
  `xmit-hash-policy {layer2 | layer3+4}

To define the bond operating mode:

set bonding group <BondID> mode <mode> [option]

Example:

set bonding group 777 mode xor xmit-hash-policy layer3+4
Defining Interfaces

A bond interface typically contains between two and eight slave interfaces. This section shows how to add and remove a slave interface. The slave interface must not have IP addresses assigned to it.

To add a slave interface to a bond:

```
add bonding group <bondID> interface <IFName>
```

Example:
```
add bonding group 777 interface eth4
```

**Note** - Do not change the bond state manually. This is done automatically by the bonding driver.

To delete a slave interface from a bond:

```
delete bonding group <bondID> interface <IFName>
```

Example:
```
delete bonding group 777 interface eth4
```

**Note** - You must delete all non-primary slave interfaces before you remove the primary slave interface.

Defining the Primary Slave Interface

With the Active-Backup operating mode, the system automatically fails over to the primary slave interface, if available. If the primary interface is not available, the system fails over to a different slave interface. By default, the first slave interface that you define is the primary interface. You must define the slave interfaces and set the operating mode as Active-Backup before doing this procedure.

**Note** - You must delete all non-primary slave interfaces before you remove the primary slave interface.

To define the primary slave interface:

```
set bonding group <bondID> mode active-backup primary <IFName>
```

Example
```
add bonding group 777 interface eth4
set bonding group 777 mode active-backup primary eth4
```

Defining the Media Monitoring Interval

This sets the frequency of requests sent to the Media Independent Interface (MII) to confirm that a slave interface is up. The valid range is 1-5000 ms. The default is 100 ms.

To configure the monitoring interval:

```
set bonding group <bondID> mii-interval <ms>
```

Example:
```
set bonding group 777 mii-interval 500
```
To disable monitoring:
```
set bonding group <bondID> mii-interval 0
```

**Defining the UP and Down Delay Times**

This parameter defines the waiting time, in milliseconds, to confirm the slave interface status before taking the specified action. Valid values are 0 to 5000 ms. The default is 200 ms.

To configure the UP and Down delay times:
```
set bonding group <bondID> down-delay <ms>
set bonding group <bondID> up-delay <ms>
```
Example:
```
set bonding group 777 down-delay 500
```

**Defining Load Sharing Parameters**

When using Load Sharing modes (XOR or 802.3ad), you can configure these parameters:

- **LACP Rate** - Set the Link Aggregation Control Protocol packet transmission rate. Valid values are slow (every 30 seconds) and fast (every 1 second).

- **Transmit Hash Policy** (802.3ad only) - Set the algorithm for interface selection according to the specified TCP/IP layer. Valid values are layer2 (uses XOR of the physical interface MAC address) and layer3+4 (uses upper layer protocol data).

To set the LACP rate:
```
set bonding group <bondID> lACP-rate {slow | fast}
```
Example: set bonding group 777 mode 8023AD lACP-rate slow

To set the Transmit Hash Policy:
```
set bonding group <bondID> xmit-hash-policy <layer>
```
Example: set bonding group 777 mode xor xmit-hash-policy layer2

**Making Sure that Link Aggregation is Working**

To make sure that a Link Aggregation is working for a bond interface, run this command in expert mode:
```
cat /proc/net/bonding/<bondID>
```
Example with output:
```
cat /proc/net/bonding/bond666
Ethernet Channel Bonding Driver: v3.2.4 (January 28, 2008)
Bonding Mode: fault-tolerance (active-backup)
Primary Slave: None
Currently Active Slave: eth2
MII Status: up
MII Polling Interval (ms): 100
Up Delay (ms): 100
Down Delay (ms): 200
Slave Interface: eth2
```
MII Status: up
Link Failure Count: 2
Permanent HW addr: 00:50:56:94:11:de
Optimizing VSX

In This Section:

- QoS Enforcement ................................................................. 140
- Monitoring Memory Resources ........................................... 148
- VSX CPU Monitoring Commands ........................................ 151
- SNMP Monitoring ................................................................. 154
- Configuring Jumbo Frames .................................................. 155

QoS Enforcement

QoS Enforcement for VSX provides the ability to control the network quality of service in the VSX network environment. QoS is based on the Differentiated Services architecture and allows assigning different transmission characteristics to different classes of service.

Differentiated Services is a computer networking architecture that specifies a simple, scalable and coarse-grained mechanism for classifying, managing network traffic and providing quality of service (QoS) guarantees on modern IP networks. Differential services can, for example, be used to provide low-latency, guaranteed service (GS) to critical network traffic such as voice or video while providing simple best-effort traffic guarantees to non-critical services such as web traffic or file transfers.

The major characteristics that are controllable by QoS are latency and bandwidth allocation. QoS is designed to provide QoS functionality with minimal impact on performance. QoS works seamlessly with Check Point Performance Pack.

The VSX network usually includes various types of traffic such as:

- Real-time traffic (e.g. VoIP) which requires low bandwidth, and is sensitive to latency (delays) and drops
- Traffic which is sensitive to latency but not to occasional drops
- High-volume, low-priority traffic which has a low sensitivity to latency and drops
- Other traffic which requires its own share of the bandwidth

Without QoS Enforcement, all these different traffic types are given equal priority on the VSX Gateway and are handled in a simple FIFO (first in-first out) manner. When the VSX Gateway is congested, all traffic types suffer the same degree of latency and drops. Also, high-volume traffic may starve other types of low-volume traffic.

With QoS, the special requirements of each traffic type can be met. For example:

- Latency-sensitive traffic will be given preference over other types of traffic
- Traffic which is sensitive to drops will suffer fewer drops than other types of traffic.
- High-volume traffic that consumes bandwidth will be limited during times of congestion.

**Note** - QoS requires the use of DiffServ-enabled routers to mark preferred traffic types with a special tag. The tag is the DSCP (DiffServ Code Point), which represents the six most significant bits of the IP header’s TOS field, as described in RFC 2474. The VSX Gateway should then be configured to give traffic with this tag the required priority.
Overview

QoS Enforcement for VSX provides the ability to control the network quality of service in the VSX network environment. QoS is based on the Differentiated Services architecture and allows assigning different transmission characteristics to different classes of service.

Differentiated Services is a computer networking architecture that specifies a simple, scalable and coarse-grained mechanism for classifying, managing network traffic and providing quality of service (QoS) guarantees on modern IP networks. Differential services can, for example, be used to provide low-latency, guaranteed service (GS) to critical network traffic such as voice or video while providing simple best-effort traffic guarantees to non-critical services such as web traffic or file transfers.

The major characteristics that are controllable by QoS are latency and bandwidth allocation. QoS is designed to provide QoS functionality with minimal impact on performance. QoS works seamlessly with Check Point Performance Pack.

The VSX network usually includes various types of traffic such as:

- Real-time traffic (e.g. VoIP) which requires low bandwidth, and is sensitive to latency (delays) and drops
- Traffic which is sensitive to latency but not to occasional drops
- High-volume, low-priority traffic which has a low sensitivity to latency and drops
- Other traffic which requires its own share of the bandwidth

Without QoS Enforcement, all these different traffic types are given equal priority on the VSX Gateway and are handled in a simple FIFO (first in-first out) manner. When the VSX Gateway is congested, all traffic types suffer the same degree of latency and drops. Also, high-volume traffic may starve other types of low-volume traffic.

With QoS, the special requirements of each traffic type can be met. For example:

- Latency-sensitive traffic will be given preference over other types of traffic
- Traffic which is sensitive to drops will suffer fewer drops than other types of traffic.
- High-volume traffic that consumes bandwidth will be limited during times of congestion.

**Note** - QoS requires the use of DiffServ-enabled routers to mark preferred traffic types with a special tag. The tag is the DSCP (DiffServ Code Point), which represents the six most significant bits of the IP header’s TOS field, as described in RFC 2474. The VSX Gateway should then be configured to give traffic with this tag the required priority.

Architecture

Three major aspects of the QoS architecture are:

- Differentiated Services support
- Inbound prioritization
- Policy with a global scope
**Differentiated Services Support**

QoS provides basic support for Differentiated Services, an architecture for specifying and controlling network traffic by class so that certain types of traffic receive priority over others. The differentiated services architecture PHB’s (per-hop behaviors).

When marked packets arrive to the VSX machine, they are classified and prioritized according to their DSCP (differential services code-point) values. To enhance performance, QoS does not mark packets with DSCP and does not change their Type of Service (ToS) values. QoS instead relies on peripheral devices (namely routers) to mark packets with the appropriate ToS value.

**Inbound Prioritization**

While Differentiated Services support in routers is usually performed on outbound traffic, QoS for VSX prioritizes traffic on the inbound side because, in VSX deployments, QoS is primarily governed by system resources, namely the CPU, and not by network bandwidth.

To prevent the VSX machine from becoming a bottleneck in the network, prioritization is enforced when packets arrive at the VSX machine, and before CPU processing is assigned.

Inbound prioritization allows an earlier control on the loss and delay rate.

**Policy with Global Scope**

To minimize the impact of QoS functionality on performance, QoS is not performed on a per interface basis, but for the entire system. This means that a certain class of service will apply to all traffic entering the VSX Gateway or cluster, regardless of the specific interface from which the traffic originates.

*Note* - On multiple-CPU machines, enforcement is not performed system-wide, but executed per-CPU. This means that global enforcement is done separately on traffic processed by each CPU.

**QoS Features**

Two main features of QoS are:

- Resource allocation
- Latency control

**Resource Allocation**

System resources are allocated by assigning different weights to different classes of service. A weight is the relative portion of the available resources allocated to a class. Allocating resources according to weights ensures full utilization of the line even if a specific class is not using all of its resources. In such a case, the remaining resources are divided among the remaining classes in accordance with their relative weights.

**Latency**

For some types of traffic, such as voice and video, it is necessary to minimize the latency (delay) of packets. Latency is controlled by defining special LLQ (low-latency queuing) classes. These classes are handled in a strict priority manner. LLQ packets are handled immediately upon arrival, and before packets that do not belong to LLQ classes.
QoS supports multiple LLQ classes. In some cases, it may be necessary to define more than one Low Latency class, for example when different types of traffic have a different sensitivity to delays. In such cases, a class with the higher sensitivity to delay receives a higher priority than a class with the lower sensitivity.

**Note** - When LLQ classes are used, it is assumed that the expected traffic will not exceed a relatively small amount of the available resources. Although QoS does not allow LLQ traffic to starve non-LLQ traffic, too much LLQ traffic reduces overall network quality of service and prevents efficient management of weighted resources.

**WRED**

RED (Random Early Drop) is a congestion avoidance mechanism for detecting and preventing congestions. It takes advantage of TCP’s congestion control mechanism by randomly dropping packets during periods of congestion. This causes TCP senders to slow down their transmission, thus preventing high congestion.

QoS implements WRED (Weighted RED) in which packets are dropped according to their priority. WRED mostly affects traffic which is of low priority and which exceeds its weight.

**QoS Management**

To manage the network quality of service it is necessary to create and install a QoS policy. The QoS policy consists of a list of up to 15 classes of service. Each class is assigned certain traffic characteristics and DSCP values.

The QoS policy is managed using the `cpqos` command.

**Class of Service Definitions**

The definition of a class of service includes the following:

- **Name.** The class name is a unique identifier which identifies the class during configuration and when presenting statistics
- **Type.** There are two types of classes, LLQ and regular classes. Regular classes are non-LLQ classes which can be assigned a weight value.
- **Priority.** Each class is assigned a unique priority value between 1 and 15. The priority value is effective in prioritizing LLQ classes and during congestion, when drops occur.
- **Weight value.** Each class is assigned a specific weight value
- **One or more DSCP values.** The Differentiated Services code point

**Priority and LLQs**

If there are multiple LLQ classes, packets are handled in a strict priority-based manner. Packets from a class with a higher priority are handled before packets with a lower priority class.

**Priority and Drop Precedence**

Priority also determines the probability of drops. A class with a lower priority has a higher drop precedence during times of congestion.
The class priority is not the only factor that determines if drops occur. Other factors affect drops, for example if the class is LLQ or if the class exceeds its assigned resource allocation.

LLQ’s are not immune to drops. Although LLQ’s are processed as soon as they arrive (and thus have a lower drop rate), drops may occur if there are many LLQ classes or if a large portion of the incoming traffic is LLQ.

### QoS Configuration

All user interactions with the QoS module are performed with the `cpqos` command.

#### The `cpqos` Command

<table>
<thead>
<tr>
<th><code>cpqos</code></th>
<th>- Manage the network quality of service.</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cpqos install</code></td>
<td>- Install the QoS policy.</td>
</tr>
<tr>
<td><code>cpqos uninstall</code></td>
<td>- Uninstall the QoS policy.</td>
</tr>
<tr>
<td><code>cpqos status</code></td>
<td>- Check if policy is installed.</td>
</tr>
<tr>
<td><code>cpqos class show [-b]</code></td>
<td>- Show the QoS policy. [-b] display dscp in binary numbers.</td>
</tr>
</tbody>
</table>
| `cpqos class add <name> prio <val> type <llq|reg> [weight <val>] dscp <val[,val2[,val3...]]>` | - Add new class with specified name 

- **name** Unique name for the class  
- **priority** Value between 1 and 15. A low value indicates a higher priority  
- **type** "llq" for low-latency classes or "reg" for regular, weighted classes  
- **weight** This value is used only for classes of type "reg". It determines the relative portion of the resources that the class will receive in relation to other weighted classes. Valid values are between 0 and 1000.

For `cpqos`:

- All commands return a zero value for success and a non-zero value for failure
- Options and argument are case-sensitive
- Examples of various `cpqos` commands ["Sample Differentiated Services Implementation" on page 146]

### `cpqos class add`

This command adds a class with the following arguments:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Unique name for the class</td>
</tr>
<tr>
<td>priority</td>
<td>Value between 1 and 15. A low value indicates a higher priority</td>
</tr>
<tr>
<td>type</td>
<td>&quot;llq&quot; for low-latency classes or &quot;reg&quot; for regular, weighted classes</td>
</tr>
<tr>
<td>weight</td>
<td>This value is used only for classes of type &quot;reg&quot;. It determines the relative portion of the resources that the class will receive in relation to other weighted classes. Valid values are between 0 and 1000.</td>
</tr>
</tbody>
</table>
### Argument | Value
--- | ---
dscp | The DiffServ code-points assigned to the class. Multiple DSCP’s can be specified, separated by commas, with no spaces between values. Values are in decimal (not binary format) with values from 0 to 63 or “default”. There can be only one class with a “default” DSCP. The default class is used for traffic without DiffServ marking (e.g. tos=0) or traffic with DSCP values that are not assigned to any other class. If no class is used as “default”, all 64 DSCP values must be assigned to the classes. A DSCP value cannot be assigned to more than one class.

Note - Changes to the policy with cpqos class add are enforced only after the policy is installed.

cpqos class del
This command deletes the class of the specified name. Changes to the policy with cpqos class del are enforced only after the policy is installed.

cpqos class show
This command shows the classes defined in the QoS policy.

cpqos install
This command installs the previously created QoS policy. It also validates the overall integrity of the policy. Once installed, the policy remains installed even if the machine reboots.

cpqos uninstall
This command un-installs the previously installed QoS policy. If un-installed, the policy will not be installed again when the machine boots.

cpqos stats
This command shows QoS statistics. cpqos stats prints a line of statistics for each of the defined classes. Each line includes the following data columns:

<table>
<thead>
<tr>
<th>Column</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>rx</td>
<td>Number of bytes that arrived to this class since the last time statistics were presented</td>
</tr>
<tr>
<td>tx</td>
<td>Number of bytes that were transmitted from this class since the last time statistics were presented</td>
</tr>
<tr>
<td>drops</td>
<td>Number of bytes that were dropped from this class since the last time statistics were presented</td>
</tr>
</tbody>
</table>

Note:
- [-u] option shows statistics separately for each CPU
- Statistics values are byte-counts, not packet counts
- Statistics values are reset after each query.
• Statistics should be presented periodically with intervals less than 1 minute.
• It is recommended to use the `watch` command to periodically present the statistics.

**QoS Policy File**

The QoS policy file is `qos_policy.C`, located in the `$FWDIR/database` directory. The QoS policy file is created when the `cpqos` command is run for the first time. The QoS policy file should not be edited manually. Use `cpqos class add/del` to create entries. To maintain multiple QoS policies, rename `qos_policy.C` or copy it to another directory, and copy it back to `$FWDIR/database/qos_policy.C` when the policy needs to be enforced.

**QoS Default Configuration**

Default QoS configuration is set to “uninstall” (e.g. not enforced). Calling `cpqos install` or `cpqos uninstall` sets the default configuration after boot.

**Sample Differentiated Services Implementation**

This section presents a sample differentiated services implementation. It includes examples for configuration, monitoring and statistics.

**Sample Traffic Types**

<table>
<thead>
<tr>
<th>Traffic Type</th>
<th>Meaning...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diamond</td>
<td>Real-time traffic (e.g. VOIP) which requires little bandwidth and is sensitive to latency and drops. This traffic is usually assigned to the EF (Expedited-Forwarding) PHB (Per Hop Behavior).</td>
</tr>
<tr>
<td>Platinum</td>
<td>Real-time traffic with low bandwidth requirements that is less sensitive to latency and drops than Diamond.</td>
</tr>
<tr>
<td>Gold</td>
<td>Traffic which is sensitive to drops</td>
</tr>
<tr>
<td>Silver</td>
<td>Traffic which is less sensitive to drops than Gold.</td>
</tr>
<tr>
<td>Bronze</td>
<td>Various types of traffic which require resource allocation. This traffic is usually assigned to the Best-Effort PHB.</td>
</tr>
<tr>
<td>Copper</td>
<td>High-volume traffic with a tendency to consume bandwidth</td>
</tr>
</tbody>
</table>

**Configuration Guidelines**

Your QoS policy should apply these guidelines:

• Diamond and Platinum classes should be defined as LLQ so they will have a lower latency then other classes
• Diamond should receive a higher priority than Platinum so it have even less latency and drops
• Gold should receive a higher priority than Silver so it will have fewer drops
• Copper resource consumption should be limited to about 10% of the available resource during periods of congestion
• Other traffic should receive about 45% of bandwidth when the traffic load is high
Configuration Examples

This examples of the `cpqos class add` command creates classes for traffic of various types:

```plaintext
cpqos class add Diamond type 11q prio 1 dscp 46
cpqos class add Platinum type 11q prio 2 dscp 32
cpqos class add Gold type reg prio 3 weight 100 dscp 26
cpqos class add Silver type reg prio 4 weight 100 dscp 28
cpqos class add Bronze type reg prio 5 weight 200 dscp default
cpqos class add Copper type reg prio 15 weight 50 dscp 10,12,14
```

Monitoring example shows previously defined classes:

```
[Expert@cpmodule:0]# cpqos class show
class: Diamond
  priority: 1
  type: 11q
  weight: 0
  DSCPs: 46

class: Platinum
  priority: 2
  type: 11q
  weight: 0
  DSCPs: 32

class: Gold
  priority: 3
  type: reg
  weight: 100
  DSCPs: 26

class: Silver
  priority: 4
  type: 11q
  weight: 100
  DSCPs: 28

class: Bronze
  priority: 5
  type: 11q
  weight: 200
  DSCPs: default

class: Copper
  priority: 15
  type: reg
  weight: 50
  DSCPs: 10,12,14
```

Statistics example shows statistics for the previously defined classes:

<table>
<thead>
<tr>
<th>class</th>
<th>priority</th>
<th>type</th>
<th>weight</th>
<th>rx</th>
<th>tx</th>
<th>drops</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diamond</td>
<td>1</td>
<td>11q</td>
<td>0</td>
<td>2775</td>
<td>2650</td>
<td>0</td>
</tr>
</tbody>
</table>

| Platinum | 2 | llq | 0 | 1024 | 1020 | 105 |
| Gold | 3 | reg | 100 | 1775015 | 1773805 | 205 |
| Silver | 4 | reg | 100 | 1862437 | 1862336 | 550 |
| Bronze | 5 | reg | 200 | 3370033 | 2955120 | 3147 |
| Copper | 15 | reg | 50 | 1862437 | 762336 | 100689 |

From this statistical output, it is apparent that:

- In the Diamond class there were no drops.
- In the Platinum class there were a few drops, even though less traffic arrived classed as Platinum than did as Diamond.
- In the Gold class there were fewer drops than from the Silver class.
- In the Bronze class there were twice as many bytes transmitted than in the Silver and Gold classes, and four times as many bytes than there were in the Copper class.
- Most packets in the Copper class were dropped.

### Monitoring Memory Resources

Use the `fw vsx mstat` command to monitor the memory the VSX Gateway uses. The command shows an overview of the memory that the system and each virtual device is using. These are the global memory resources that are shown:

- **Memory Total** - Total physical memory on the VSX Gateway.
- **Memory Free** - Available physical memory.
- **Swap Total** - Total of swap memory.
- **Swap Free** - Available swap memory.
- **Swap-in rate** - Total memory swaps per second.

The virtual devices are listed according to the VSIDs. Run `vsx stat -v` to show the VSID for the virtual devices.

You must be in expert mode to run the `fw vsx mstat` command. After you enable memory resource monitoring, it is necessary to reboot the VSX Gateway to use the feature.

### Managing `fw vsx mstat`

Use the `fw vsx mstat` command to enable or disable memory resource monitoring on the VSX Gateway.

#### Syntax

```
fw vsx mstat {enable|disable|status}
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>enable</td>
<td>Enables memory resource monitoring. Reboot the VSX Gateway.</td>
</tr>
<tr>
<td>disable</td>
<td>Disables memory resource monitoring.</td>
</tr>
<tr>
<td>status</td>
<td>Shows if memory resource monitoring is enabled or disabled.</td>
</tr>
</tbody>
</table>
Example

fw vsx mstat disable

Output

VSX memory resource control status: disabled

Memory Resources for Each virtual device

Use the `fw vsx mstat` command to show the memory that each virtual device uses. Use the `--vs` parameter to show only some of the virtual devices.

You can also use these parameters for more data:

- `unit` - Change the memory measurement unit shown in the command output.
- `sort` - Sort the results according to the virtual devices that use the most memory and limit the display to the specified number of results.

Syntax

`fw vsx mstat [-vs <VSID>] [unit <measure>] [sort <top>]`

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-vs</code></td>
<td>Shows the memory usage of the specified virtual devices.</td>
</tr>
<tr>
<td><code>&lt;VSID&gt;</code></td>
<td>The ID of the virtual device. To show multiple devices:</td>
</tr>
<tr>
<td></td>
<td>• Put a space between each VSID: <code>-vs 1 3 5</code></td>
</tr>
<tr>
<td></td>
<td>• List a range of VSIDs: <code>-vs 1-4</code></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: You can combine VSID ranges together with single VSIDs</td>
</tr>
<tr>
<td><code>unit</code></td>
<td>Change the memory measurement unit shown in the command output.</td>
</tr>
<tr>
<td><code>&lt;measure&gt;</code></td>
<td>The memory measurement unit. The default value is megabytes.</td>
</tr>
<tr>
<td></td>
<td>The values are:</td>
</tr>
<tr>
<td></td>
<td>• B - bytes</td>
</tr>
<tr>
<td></td>
<td>• K, KB - kilobytes</td>
</tr>
<tr>
<td></td>
<td>• M, MB - megabytes (default)</td>
</tr>
<tr>
<td></td>
<td>• G, GB - gigabytes</td>
</tr>
<tr>
<td><code>sort</code></td>
<td>Sort the results according to the virtual devices that use the most memory.</td>
</tr>
<tr>
<td><code>&lt;top&gt;</code></td>
<td>Enter the maximum number of virtual devices to be shown. Only those virtual devices that use the most memory are shown. Use <code>all</code> to show all virtual devices.</td>
</tr>
</tbody>
</table>

Example

```
fw vsx mstat -vs 0 1 3 5-8 unit MB sort 5
fw vsx mstat sort 5
```
Output [Both examples show the same results]

<table>
<thead>
<tr>
<th>VSX Memory Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Total: 997.22 MB</td>
</tr>
<tr>
<td>Memory Free: 232.56 MB</td>
</tr>
<tr>
<td>Swap Total: 2047.34 MB</td>
</tr>
<tr>
<td>Swap Free: 2047.16 MB</td>
</tr>
<tr>
<td>Swap-in rate: 0.00 MB</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VSID</th>
<th>Memory Consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>133.50 MB</td>
</tr>
<tr>
<td>8</td>
<td>92.41 MB</td>
</tr>
<tr>
<td>3</td>
<td>43.81 MB</td>
</tr>
<tr>
<td>6</td>
<td>42.47 MB</td>
</tr>
<tr>
<td>1</td>
<td>42.47 MB</td>
</tr>
</tbody>
</table>

Configuring Swap-in Sample Rate

The swap-in rate measures how much memory per second that the system swaps-in from disk. You can configure how often the system calculates the swap-in rate. For example, a sample rate of 5 means that the system calculates the swap-in rate every five minutes.

Syntax

fw vsx mstat swap <minutes>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>swap</td>
<td>Configures the swap-in sample rate for the system.</td>
</tr>
<tr>
<td>&lt;minutes&gt;</td>
<td>Number of minutes that the system measures memory swaps to determine the swap-in rate. Only integers are valid values. The default swap-in sample rate is 10.</td>
</tr>
</tbody>
</table>

Example

fw vsx mstat swap 5

Output

Swap-in sample rate was changed successfully to 5 minutes.

Comments

Swap-in sample rate is a system wide Linux setting. When you change the value for memory monitoring, all the swap-in rates are calculated according to the new value.

When you enable the monitoring memory resources feature, the swap-in rate setting is saved. When you disable the feature, the system restores the saved setting.

Using Debug Mode

Use the debug parameter to show more data about the memory that the VSX Gateway uses. You cannot use the -vs, unit and sort parameters in debug mode. The memory is shown in kilobytes.
Syntax

fw vsx mstat debug

Output

VSX Memory Status
==================
Memory Total: 1021152.00 KB
Memory Free: 324788.00 KB
Swap Total: 2096472.00 KB
Swap Free: 2096404.00 KB
Swap-in rate: 375.34 KB

<table>
<thead>
<tr>
<th>VSID</th>
<th>Private_Clean</th>
<th>Private_Dirty</th>
<th>DispatcherGConn</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>13544.00 KB</td>
<td>144268.00 KB</td>
<td>0.00 KB</td>
</tr>
<tr>
<td>1</td>
<td>1740.00 KB</td>
<td>46276.00 KB</td>
<td>0.00 KB</td>
</tr>
<tr>
<td>2</td>
<td>1720.00 KB</td>
<td>46868.00 KB</td>
<td>0.00 KB</td>
</tr>
<tr>
<td>3</td>
<td>1720.00 KB</td>
<td>46644.00 KB</td>
<td>0.00 KB</td>
</tr>
<tr>
<td>4</td>
<td>1712.00 KB</td>
<td>45144.00 KB</td>
<td>0.00 KB</td>
</tr>
<tr>
<td>5</td>
<td>1712.00 KB</td>
<td>45836.00 KB</td>
<td>0.00 KB</td>
</tr>
<tr>
<td>6</td>
<td>1720.00 KB</td>
<td>45000.00 KB</td>
<td>0.00 KB</td>
</tr>
<tr>
<td>7</td>
<td>1720.00 KB</td>
<td>45044.00 KB</td>
<td>0.00 KB</td>
</tr>
</tbody>
</table>

Comments

By default the debug parameter shows these memory fields:

Private_Clean  - Clean private pages. /proc/[pid]/smaps

Private_Dirty  - Dirty private pages. /proc/[pid]/smaps

DispatcherHTab - Hash table for each Virtual System.

DispatcherGConn - Global connections for each Virtual System.

SecureXL  - SecureXL memory each Virtual System uses.

VSX CPU Monitoring Commands

Use the fw vsx resctrl commands to monitor the CPU resources on a VSX Gateway. You can also see real-time statistics on the current and average CPU consumption by the virtual devices.

fw vsx resctrl load_configuration

Description

Initializes Resource Control. This command uses the contents of the resctrl file to configure CPU resource Control.

Syntax

fw vsx resctrl load_configuration

Output

Loading Resource Control configuration from $FWDIR/conf/resctrl
Resource Control Monitor is disabled
Resource Control monitoring configuration successfully read.
Comment
After you run `fw vsx resctrl load_configuration`, the commands in the `resctrl` file enable or disable Resource Control monitoring.

`fw vsx resctrl monitor`

Description
Configures the Resource Monitor and shows its current status. This command overrides the settings in the Resource Control configuration file, but does not survive reboot.

Syntax
```
fw vsx resctrl monitor {enable | disable | show}
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>enable</td>
<td>Enables Resource Monitor</td>
</tr>
<tr>
<td>disable</td>
<td>Disables Resource Monitor</td>
</tr>
<tr>
<td>show</td>
<td>Displays whether Resource Monitor is enabled or disabled</td>
</tr>
</tbody>
</table>

Example
```
fw vsx resctrl monitor enable
```

Output
```
Resource Control Monitor is enabled
```

`fw vsx resctrl reset`

Description
Resets the Resource Control monitoring statistics.

Syntax
```
fw vsx resctrl reset
```

`fw vsx resctrl stat`

Description
Shows the percentage of the total CPU resources that each virtual device uses.

Syntax
```
fw vsx resctrl [-u | -d | -d -q] stat
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-u</td>
<td>Displays information per CPU [SMP only].</td>
</tr>
</tbody>
</table>
-d Displays 24 hours of statistics. These statistics are only available after 24 hours of monitoring.

-d -q Same as -d parameter without details for each CPU. Shows an average of all the CPUs for each virtual device.

Example

`fw vsx resctrl -d -q stat`

Output

Virtual Systems CPU Usage Statistics
=====================================
Number of CPUs/Hyper-threading: 4
Monitoring active time: 14s

<table>
<thead>
<tr>
<th>ID Name</th>
<th>1sec</th>
<th>10sec</th>
<th>1min</th>
<th>1hr</th>
<th>24hr*</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 VSX2</td>
<td>0.11</td>
<td>0.06</td>
<td>0.08</td>
<td>0.07</td>
<td>0.01</td>
</tr>
<tr>
<td>1 VSX2 vs1</td>
<td>15.80</td>
<td>21.57</td>
<td>21.75</td>
<td>22.28</td>
<td>1.94</td>
</tr>
<tr>
<td>2 VSX2 vs2</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>3 VSX2 vs2</td>
<td>16.91</td>
<td>22.57</td>
<td>22.77</td>
<td>23.09</td>
<td>2.01</td>
</tr>
</tbody>
</table>

Total Virtual Devices CPU Use: 32.82 44.20 44.60 45.44 3.96

Sample Output fw vsx resctrl -u

Virtual Systems CPU Usage Statistics [%]
========================================
Number of CPUs: 4
Monitoring active time: 9m 4s

<table>
<thead>
<tr>
<th>ID Name</th>
<th>CPU</th>
<th>1sec</th>
<th>10sec</th>
<th>1min</th>
<th>1hr*</th>
<th>24hr*</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 vsxb</td>
<td>0</td>
<td>0.30</td>
<td>0.44</td>
<td>0.48</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>1 vsxb</td>
<td>1</td>
<td>12.50</td>
<td>1.33</td>
<td>0.48</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>2 vsxb</td>
<td>2</td>
<td>3.40</td>
<td>1.36</td>
<td>5.25</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>3 vsxb</td>
<td>3</td>
<td>0.20</td>
<td>0.24</td>
<td>0.41</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Avg.</td>
<td></td>
<td>4.10</td>
<td>0.84</td>
<td>1.65</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>1 vs1</td>
<td>0</td>
<td>0.00</td>
<td>0.02</td>
<td>0.11</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>1 vs1</td>
<td>1</td>
<td>0.00</td>
<td>0.01</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>2 vs1</td>
<td>2</td>
<td>0.20</td>
<td>0.04</td>
<td>0.11</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>3 vs1</td>
<td>3</td>
<td>0.00</td>
<td>0.01</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Avg.</td>
<td></td>
<td>0.05</td>
<td>0.02</td>
<td>0.06</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Total Virtual Devices CPU Use: 0.30 0.46 0.59 0.00 0.00

Comments

- For systems with more than one CPU, time is an average for all CPUs. To see the usage for each virtual device per CPU, run `vsx resctrl -u stat`
Total Virtual System CPU Usage includes the total for all virtual devices: Virtual Routers, Virtual Switches, Virtual Systems and the VSX Gateway.

**SNMP Monitoring**

You can only send SNMP traps to VS0.

For more about using SNMP, see SNMP in the *R76 Gaia Administration Guide*

**VSX SNMP Modes**

There are two modes of SNMP monitoring that you can use with VSX:

- Default mode
- VS mode

VS mode lets you monitor all of the Virtual Systems in the VSX Gateway. Default mode only monitors VS0.

**Note** - When SNMP VS mode is enabled, traps from Virtual Systems are not supported.

**Supported SNMP Versions**

The VS mode uses SNMP version 3 to query the Virtual Systems. You can run remote SNMP queries on Virtual Systems in the VSX Gateway.

For systems that only support SNMP versions 1 and 2:

- You cannot run remote SNMP queries for each Virtual System. You can only run a remote SNMP query on VS0.
- You can use the CLI to change the Virtual System context and then run a local SNMP query on a Virtual System.

**Enabling VS Mode**

To enable VS mode on the VSX Gateway:

1. Create an SNMP v3 user.
2. Enable VS mode.
3. Start the SNMP agent.

Sample commands:

```
> add snmp usm user admin security-level authNoPriv auth-pass-phrase abcd1234
> set snmp mode vs
> set snmp agent on
```

**Note** - If you run SNMP in the VS mode, you must run the `snmpwalk` command and set the `-n` switch to `ctxname_vsid<vsid>`.  

**Example:**  

```
snmpwalk -n ctxname_vsid2 -v 3 -l authNoPriv -u admin -A abcd1234 192.2.2.2 ifDesc
```
Sample SNMP VS Mode Configuration

Sample remote query of interfaces for VS2:
> snmpwalk -n ctxname_vsid2 -v 3 -l authNoPriv -u admin -A abdc1234 ifDescr

Sample local query of interfaces for VS2:
> vsenv 2
> snmpwalk -v 2c -c public localhost ifDescr

Note - You must enable VSX Resource Control Monitoring in order to see information about CPU usage per Virtual System.

Configuring Jumbo Frames

VSX supports Jumbo Frames and lets you configure up to the maximum MTU of the NIC driver.

Jumbo Frames on a Virtual System

Configure a Virtual System to enable Jumbo Frames on these interfaces types:

- Physical interface
- Bond interface
- VLAN interface

When you configure the MTU of a bond interface, the MTU of all the slave interfaces are automatically changed to the new value.

Configuring the MTU on Warp Links

- To change the MTU of Warp Links that connect to a Virtual Switch, configure the Virtual Switch settings.
- You can change the MTU of Warp Links that connect to a Virtual Router in the Virtual System settings.

Configuring the MTU on VLANs

- When you increase the MTU of a VLAN trunk, the value of the VLAN interfaces are not changed.
- The MTU of the VLAN trunk cannot be less than the value of a VLAN interface. If you increase the MTU of a VLAN interface to be larger than the VLAN trunk, the MTU of the VLAN trunk is increased to the new value for the VLAN interface.

To configure Jumbo Frames on a Virtual System:

1. Open SmartDashboard.
2. From the Network Objects tree, right-click the Virtual System and select Edit. The General Properties window opens.
3. Click Topology.
4. Select the interface and click Edit. The Interface Properties window opens.
5. Configure the **MTU** for the interface.
6. Click **OK**.

**Jumbo Frames on a Virtual Switch**

Configure the MTU of a Virtual Switch to enable Jumbo Frames on the Virtual Systems that are connected to the Virtual Switch. When you configure the MTU of the Virtual Switch, all the related Warp Links and interfaces are automatically changed to the new value.

You cannot configure the MTU of a Warp Link from the Virtual System.

To configure Jumbo Frames on a Virtual Switch:
1. Open SmartDashboard.
2. From the **Network Objects** tree, right-click the Virtual Switch and select **Edit**.
   The **General Properties** window opens.
3. Click **Topology**.
4. Configure the **MTU** for the Virtual Switch.
5. Click **OK**.

**Jumbo Frames on a Virtual Router**

Configure the MTU of a Virtual Router to enable Jumbo Frames on the interfaces. To change the MTU of Warp Links, configure the settings of the Virtual System.

To configure Jumbo Frames on a Virtual Router:
1. Open SmartDashboard.
2. From the **Network Objects** tree, right-click the Virtual Router and select **Edit**.
   The **General Properties** window opens.
3. Click **Topology**.
4. Select the interface and click **Edit**.
   The **Interface Properties** window opens.
5. Configure the **MTU** for the interface.
6. Click **OK**.

**Jumbo Frames on a Virtual System in Bridge Mode**

Configure the MTU of a Virtual System in Bridge Mode to enable Jumbo Frames on the interfaces.

To configure Jumbo Frames on a Virtual System in Bridge Mode:
1. Open SmartDashboard.
2. From the **Network Objects** tree, right-click the Virtual System and select **Edit**.
   The **General Properties** window opens.
3. Click **Topology**.
4. Configure the **MTU** for the interface.
5. Click **OK**.
VSX Diagnostics and Troubleshooting

**In This Section:**
- Introduction ................................................................................................................. 157
- General Troubleshooting Steps ................................................................................. 157
- Troubleshooting Specific Problems .......................................................................... 158

This chapter presents basic diagnostic and troubleshooting procedures that should be followed in the event you encountering a problem while working with VSX. This diagnostic routine will assist you in determining the source of the problem. This chapter presents several known issues and their solutions.

Most problems are caused by configuration errors occurring during the process of defining VSX Gateway, clusters and/or virtual devices. Another common source of problems involves networking and connectivity issues affecting VSX behavior. These problems are listed according to the order in which you will likely encounter them. Before reading and following a certain workaround, make sure you’ve read all the previous workarounds, and that those steps in the configuration were successful.

In some of the cases, one initial problem can cause problems in later stages of the configuration. For that reason, it is important to find the root of the problem when you are trying to understand what went wrong.

**Introduction**

This chapter presents basic diagnostic and troubleshooting procedures that should be followed in the event you encountering a problem while working with VSX. This diagnostic routine will assist you in determining the source of the problem. This chapter presents several known issues and their solutions.

Most problems are caused by configuration errors occurring during the process of defining VSX Gateway, clusters and/or virtual devices. Another common source of problems involves networking and connectivity issues affecting VSX behavior. These problems are listed according to the order in which you will likely encounter them. Before reading and following a certain workaround, make sure you’ve read all the previous workarounds, and that those steps in the configuration were successful.

In some of the cases, one initial problem can cause problems in later stages of the configuration. For that reason, it is important to find the root of the problem when you are trying to understand what went wrong.

**General Troubleshooting Steps**

If you suspect that there is a problem with your VSX configuration, there are several diagnostic procedures that you can follow to determine the source. These procedures utilize various commands documented in the Command Line section [“Command Line Reference” on page 162].

1. **Perform a basic configuration check for each gateway or cluster member by running the `fw vsx stat -v` command.** The output will allow you to:
a) Account for all Virtual Systems and verify that none are missing from the configuration.
b) Verify that all virtual devices are active
c) Verify that the correct security policy is installed for each Virtual System
d) Verify the SIC trust has been established with the management server

2. Run the `cplic print` command on each VSX Gateway, cluster member and management server to verify that you have the appropriate licenses installed.

3. Run the `cphaprojb stat` command on each cluster member to verify its status. If a member is listed with a status other than Active, Standby, or Backup, refer to the “Troubleshooting” chapter in the R76 ClusterXL Administration Guide for additional troubleshooting assistance.

4. If you suspect that a Virtual System is experiencing connectivity problems, perform the following steps:
   a) Run: `vsenv` to set the context to the appropriate Virtual System.
   b) Run `fw getifs` to display the interface list for the Virtual System.
   c) Examine connectivity status using standard operating system commands and tools such as `ping`, `traceroute`, `tcpdump`, `ip route`, `ftp` etc. Some of these run according to context (i.e. routing, source and destination IP addresses).
      You can also execute the `ip route` and `ip link` commands.
      If these tests indicate that all interfaces and routers have connectivity, and appear to be functioning correctly, you should monitor the passage of packets through the system.

5. Execute the `fw monitor -v <vsid>` commands to capture details of packets at multiple points. This may return multiple reports on the same packet as it passes various capture points. This command does not report on Virtual Routers, except for packets destined to an external Virtual Router.
   **Note** - The Performance Pack may have an adverse effect on the capabilities of the `fw monitor` command.

6. Execute the `tcpdump` command to display transmitted or received packets for specific interfaces, including Warp interfaces. This often provides valuable clues for resolving connectivity issues.

## Troubleshooting Specific Problems

### Cannot Establish SIC Trust for Gateway or Cluster

When creating a VSX Gateway or cluster, you cannot establish SIC trust. SmartDashboard gives an error message:

**Certificate cannot be pushed. Connection error with wait agent.**
### Possible Causes

**Check that you have network connectivity between the gateway and the Security Gateway or Domain Management Server by pinging from the VSX system [A ping from the Domain Management Server/Security Management to the VSX system will not work because of the default security policy installed on the VSX Gateway/cluster.]**

Make sure the context is `vrf 0` first.

**Check that all the Check Point processes on the VSX Gateway(s) are up and running by running `cpwd_admin list` and making sure each line has a non-zero value in the PID field.**

If the gateway(s) has just rebooted, the Check Point processes might still be coming up. If this is not the case, and you are using Crossbeam X40, make sure you have executed the `application ... start` command. (For more information refer to the Crossbeam documentation.)

**Check that the CPD process is listening to the trust establishment port.**

Run `netstat -an | grep 18211` on the VSX Gateway(s), and make sure that output looks like this:

```plaintext
tcp 0 0 0.0.0.0:18211 0.0.0.0:* LISTEN
```

### SIC Trust Problems with New virtual devices

When creating a new Virtual System, Virtual Router or Virtual Switch, you cannot establish SIC trust.

<table>
<thead>
<tr>
<th>Possible Causes</th>
<th>How to Resolve</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Time or time zone mismatch between the management and the gateway.</strong></td>
<td>Change the time, date and time zone on the management and/or the gateway so that their UTC/GMT times match. Refer to your operating system documentation for the exact commands needed to accomplish this.</td>
</tr>
<tr>
<td>For proper SIC operation, the time, date and time zone must be synchronized between the management server and Gateways/ cluster members.</td>
<td></td>
</tr>
<tr>
<td>Execute the <code>/bin/date -u</code> command on all machines, to obtain the correct UTC/GMT time. The machines can be in different time zones, as long as their UTC/GMT times match.</td>
<td></td>
</tr>
</tbody>
</table>

### Re-establishing SIC Trust with virtual devices

In the event that you encounter connectivity problems due to the loss of SIC trust for a specific virtual device, you can use the following procedure to manually re-establish trust.
To manually re-establish SIC Trust with virtual devices:

1. Execute the following command from the VSX Gateway command line (In the expert mode):
   ```
   vsx sic reset <vsid>.
   
   vsid: Identification number of the virtual device
   ```

2. Execute the following command(s) on the management server:
   a) 
   ```
   # mdsenv <target_domain_name>
   
   (Multi-Domain Security Management only)
   ```
   b) 
   ```
   # cpca_client revoke_cert -n <vs_sic_name>
   
   vs_sic_name: virtual device SIC name. To determine the SIC name, run guidbedit.exe and search for the sic_name attribute on the virtual device network object.
   ```

3. In SmartDashboard, open the virtual device object and click **OK**. This action creates a new SIC certificate for the virtual device and saves it on the VSX Gateway.

### Install Policy Error Using VSX Creation Wizard

After completing the VSX creation wizard, a failure occurs and the following message appears in the **Operation Report** window: *Error: Default policy installation failed on VSX. Install policy manually using SmartDashboard.*

<table>
<thead>
<tr>
<th>Possible Causes</th>
<th>How to Resolve</th>
</tr>
</thead>
<tbody>
<tr>
<td>Missing or invalid license on the management server.</td>
<td>Obtain and install the appropriate licenses.</td>
</tr>
<tr>
<td>Execute <code>cplic check</code> on the management server to verify that you have the</td>
<td>Obtain a VSX and install a valid license for each VSX Gateway or cluster member.</td>
</tr>
<tr>
<td>required licenses.</td>
<td></td>
</tr>
<tr>
<td>Missing or invalid VSX Gateway / cluster licenses. Run <code>fw vsx stat</code> on all</td>
<td></td>
</tr>
<tr>
<td>Gateways, and make sure that the output says **Number of Virtual Systems</td>
<td></td>
</tr>
<tr>
<td>allowed by license:** is greater than 0.</td>
<td></td>
</tr>
<tr>
<td><strong>Time or time zone mismatch between the management and the gateway.</strong></td>
<td></td>
</tr>
<tr>
<td>For proper SIC operation, the time, date and time zone must be synchronized</td>
<td></td>
</tr>
<tr>
<td>between the management server and Gateways / cluster members.</td>
<td></td>
</tr>
<tr>
<td>Execute the <code>/bin/date -u</code> command on all machines, to obtain the correct</td>
<td>Change the time, date and time zone on the management and/or the gateways so that their UTC/GMT times match.</td>
</tr>
<tr>
<td>UTC/GMT time. The machines can be in different time zones, as long as their</td>
<td>Refer to you operating system documentation for the exact commands needed to accomplish this.</td>
</tr>
<tr>
<td>UTC/GMT times match.</td>
<td></td>
</tr>
</tbody>
</table>
## Internal Host Cannot Ping Virtual System

After defining a Virtual System with an internal VLAN interface, an internal host on that VLAN cannot ping the Virtual System internal or external IP address.

<table>
<thead>
<tr>
<th>Possible Causes</th>
<th>How to Resolve</th>
</tr>
</thead>
<tbody>
<tr>
<td>A policy allowing the communication was not installed on the Virtual System. Note that after creating a Virtual System, it has a Default Policy that blocks all traffic.</td>
<td>Install a policy on the Virtual System that enables the traffic. Check with the SmartView Tracker that the Virtual System is allowing the traffic.</td>
</tr>
<tr>
<td>There is the VLAN configuration problem on a switch, or physical cable problem.</td>
<td>Check the switch configuration. Make sure that VLAN tag configured on the switch is the same as used for the Virtual System VLAN interface. Check the cables, and make sure that you have plugged the cable from the switch to the correct port on the VSX Gateway or cluster members.</td>
</tr>
<tr>
<td>Incorrect routing on adjacent routers or hosts.</td>
<td>Check the routing tables on intermediate routers and hosts. You can use <code>tcpdump</code> on the relevant VLAN interface on the VSX Gateway or cluster member to verify that the traffic is arriving to and leaving the VSX machine.</td>
</tr>
<tr>
<td>Incorrect IP address or net mask defined on the Virtual System VLAN interface.</td>
<td>Check the IP address and the net mask assigned to the Virtual System internal VLAN interface.</td>
</tr>
</tbody>
</table>
Changing the Context

Use the set virtual-system or vsenv command to change context to a different virtual device.

- set virtual-system - Run from the CLI
- vsenv - Run from Expert mode

Syntax

set virtual-system <vsid>
vsenv <vsid>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vsid</td>
<td>ID of the virtual device.</td>
</tr>
</tbody>
</table>

Example

set virtual-system 3

Output

Context is set to vsid 3

VSX Gaia CLI Commands

This section describes how to use Gaia CLI commands on VSX Gateways. For more about Gaia CLI commands, see the R76 Gaia Administration Guide http://supportcontent.checkpoint.com/solutions?id=sk105938.

Many CLI commands for VSX are run on each context (virtual device) separately. Change the context to a different virtual device and then run the command.

Configuring Roles on virtual devices

Configure roles permissions for users on each virtual device. When you create virtual devices, they do not receive any role permissions. You can use the CLI to:
• Add a new role definition to a virtual device.
• Delete a role definition from a virtual device or all the virtual devices.

R76 VSX does not support the Gaia WebUI. You cannot configure role permissions for the WebUI.


Syntax
add rba role <name> virtual-system-access <vsid>
delete rba role <name> virtual-system-access <vsid|all>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Role name as a character string that contains letters, numbers or the underscore (_) character. The role name must start with a letter.</td>
</tr>
<tr>
<td>vsid</td>
<td>ID of the virtual device. Use commas to enter multiple virtual devices.</td>
</tr>
<tr>
<td>all</td>
<td>Deletes access permissions to all virtual devices.</td>
</tr>
</tbody>
</table>

Example
add rba role newRole domain-type System all-features
add rba role newRole virtual-system-access 0,2,3
save config

Adding Roles
Use the CLI to create and configure roles for the VSX Gateway. Assign virtual devices to a role ("Adding or Deleting virtual devices to a Role" on page 164), so that the users of that role can access them.

To create roles for a VSX Gateway:
1. Create the role. Run add rba role
2. Add virtual devices to the role. Run add rba role <name> virtual-system-access <vsid>
3. Save the role. Run save config

This is an example of creating roles.

Adding Users
Use the CLI to create new users and assign them to a role.

To add users to a role:
1. Create the user. Run add user <username> uid <user-id> homedir <home directory path>
2. Create a password for the user. Run set user <username> password
3. Add the user to a role. Run add rba user role <role name>
4. Save the user. Run save config

This is an example of creating users.
GizaVSXGW1:0> add user user1 uid 160 homedir /home/user1
WARNING Must set password and a role before user can login.
- Use 'set user USER password' to set password.
- Use 'add rba user USER roles ROLE' to set a role.
- Use 'add rba role ROLE virtual-system-access' to set allowed virtual-system role.
GizaVSXGW1:0> set user user1 password
New password:
Verify new password:
GizaVSXGW1:0> add rba user user1 roles newRole

**Adding or Deleting virtual devices to a Role**

Use the add and delete rba role commands to configure the role permissions for virtual devices.

**Syntax**

add rba role <name> virtual-system-access <vsid>
delete rba role <name> virtual-system-access <vsid|all>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Role name as a character string that contains letters, numbers or the underscore (_) character. The role name must with a letter.</td>
</tr>
<tr>
<td>vsid</td>
<td>ID of the virtual device. Use commas to enter multiple virtual devices.</td>
</tr>
<tr>
<td>all</td>
<td>Deletes access permissions to all virtual devices.</td>
</tr>
</tbody>
</table>

**Example**

add rba role newRole virtual-system-access 0,2
delete rba role newRole virtual-system-access all

**Showing virtual devices**

Use the virtual-system command to show the virtual devices on the VSX Gateway.

**Syntax**

show virtual-system all

**Output**

<table>
<thead>
<tr>
<th>Virtual Systems list</th>
</tr>
</thead>
<tbody>
<tr>
<td>VS ID</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
</tbody>
</table>

**CoreXL Commands (fw ctl affinity)**

This section explains how to use the fw ctl affinity command to set affinities in VSX. When you run this command, VSX automatically creates or updates the affinity configuration files. All affinity configurations are saved even after you reboot the VSX Gateway.
Affinity Exceptions

There are processes that are affinity exceptions and are not included in the affinity commands that you run. The $FWDIR/conf/vsaffinity_exception.conf file contains the list of processes that are affinity exceptions.

⚠️ Important - Do not add Check Point processes to this list. Doing so can make the system unstable.

You cannot set affinity to kernel threads.

Affinity Priorities

When there is a conflict between affinities, there are priorities that are used to determine which CPU affinity is used. This is the priority order from highest to lowest:

1. Firewall instance
2. Process
3. Virtual System

Setting Affinities

Run the fw ctl affinity command to set these CPU affinities:

- Firewall instance
- Process
- Virtual System

You must be in Expert mode to run the fw ctl affinity command.

Firewall Instance for each Virtual System

Set the affinity of firewall instances to one or more CPUs for each Virtual System separately.

Syntax

fw ctl affinity -s -d {-inst <instances> -cpu <cpus>|-fwkall <cores>}

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;instances&gt;</td>
<td>Number range of firewall instances that you are setting affinity. Use a dash to set a range of instances.</td>
</tr>
<tr>
<td>&lt;cpus&gt;</td>
<td>Number range of CPU processing cores that you are setting affinity. Use a dash to set a range of cores.</td>
</tr>
<tr>
<td>&lt;cores&gt;</td>
<td>Number of cores that are used for CPU affinity. You cannot use this parameter to assign specific cores to the firewall instances.</td>
</tr>
</tbody>
</table>

Example

vsev 1

fw ctl affinity -s -d -inst 0 2-4 -cpu 0-2
fw ctl affinity -s -d -fwkall 3
Output

VDevice 1: CPU 0 1 2 - set successfully

Notes
You can use this command with the -fwkall parameter from any context.

**Firewall Instance for all Virtual Systems**

Use the -fwkall parameter to set the affinity of all the firewall instances to all the Virtual Systems.

**Syntax**

```
fw ctl affinity -s -d -fwkall <cores>
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;cores&gt;</td>
<td>Number of cores that are used for CPU affinity. You cannot use this parameter to assign specific cores to the firewall instances.</td>
</tr>
</tbody>
</table>

**Example**

```
fw ctl affinity -s -d -fwkall 3
```

Output

VDevice 0-2 : CPU 3 4 5 6 7 - set successfully

Notes
You can use this command with the -fwkall parameter from any context.

**Processes**

Set the affinity of processes to one or more CPUs. You can use the -vsid parameter to set the affinity for a process to Virtual Systems in any context. If you do not use the -vsid parameter, the affinity of the current context is set.

**Syntax**

```
fw ctl affinity -s -d -pname <process> [-vsid <vsids>] -cpu <cpus>
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>process</td>
<td>Name of process that you are setting affinity.</td>
</tr>
<tr>
<td>vsids</td>
<td>Virtual System IDs that you are setting affinity for this process.</td>
</tr>
<tr>
<td></td>
<td>Use a dash to set a range of Virtual Systems.</td>
</tr>
<tr>
<td>cpus</td>
<td>Number range of CPU processing cores that you are setting affinity.</td>
</tr>
<tr>
<td></td>
<td>Use a dash to set a range of cores.</td>
</tr>
</tbody>
</table>

**Example**

```
fw ctl affinity -s -d -pname cpd -vsid 0-1 -cpu 0 2
```
Output

VDevice 0-1 : CPU 0 2 - set successfully

Virtual Systems

Set the affinity of the Virtual Systems to one or more CPUs. You can use the \(-\text{vsid}\) parameter to set affinity to the specified Virtual Systems. If you do not use the \(-\text{vsid}\) parameter, the affinity of the current Virtual System is set.

Syntax

\texttt{fw\ ctl\ affinity\ -s\ -d\ [-vsid\ <vsids>]\ -cpu\ <cpus>}

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vsids</td>
<td>Virtual System IDs that you are setting affinity. Use a dash to set a range of Virtual Systems.</td>
</tr>
<tr>
<td>cpus</td>
<td>Number range of CPU processing cores that you are setting affinity. Use a dash to set a range of cores.</td>
</tr>
</tbody>
</table>

Example

\texttt{fw\ ctl\ affinity\ -s\ -d\ -vsid\ 0-1\ -cpu\ 0 2}

Output

VDevice 0-1 : CPU 0 2 - set successfully

Affinity Monitoring

You can monitor the affinity of processes and Virtual Systems on the VSX Gateway.

Monitoring Processes

Monitor the affinity of processes on the VSX Gateway. You can use the \(-\text{vsid}\) parameter to show the affinity for a process to the specified Virtual Systems.

Syntax

\texttt{fw\ ctl\ affinity\ -l\ -x\ [-vsid\ <vsids>]\ [-flags\ [e|h|k|n|t]}

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vsids</td>
<td>Shows the affinity for processes for these Virtual System IDs. Use a dash to set a range of Virtual Systems.</td>
</tr>
<tr>
<td>e</td>
<td>Do not show processes that are affinity exceptions. Affinity exceptions are configured in the $FWDIR/conf/vsaffinity_exception.conf file.</td>
</tr>
<tr>
<td>h</td>
<td>Show CPU affinity mask in hexadecimal format.</td>
</tr>
<tr>
<td>k</td>
<td>Do not show kernel threads.</td>
</tr>
<tr>
<td>n</td>
<td>Show the process name instead of (/proc/&lt;pid&gt;/cmdline)</td>
</tr>
</tbody>
</table>
Show information about the process threads.

Example

```
fw ctl affinity -l -x -vsid 1 -flags tn
```

Output

<table>
<thead>
<tr>
<th>PID</th>
<th>VSID</th>
<th>CPU</th>
<th>SRC</th>
<th>V</th>
<th>KT</th>
<th>EXC</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>4756</td>
<td>0</td>
<td>all</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pm</td>
</tr>
<tr>
<td>4773</td>
<td>0</td>
<td>all</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>confd</td>
</tr>
<tr>
<td>4774</td>
<td>0</td>
<td>all</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>searchd</td>
</tr>
<tr>
<td>5008</td>
<td>0</td>
<td>all</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4780</td>
<td>0</td>
<td>all</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>httpd2</td>
</tr>
<tr>
<td>4781</td>
<td>0</td>
<td>all</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>monitord</td>
</tr>
<tr>
<td>24700</td>
<td>0</td>
<td>0 1 P</td>
<td></td>
<td>P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24704</td>
<td>0</td>
<td>0 1 P</td>
<td></td>
<td>P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24705</td>
<td>0</td>
<td>0 1 P</td>
<td></td>
<td>P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22800</td>
<td>0</td>
<td>all</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>mpdaemon</td>
</tr>
<tr>
<td>24523</td>
<td>0</td>
<td>all</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>fwk_forker</td>
</tr>
<tr>
<td>24525</td>
<td>0</td>
<td>all</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>fwk wd</td>
</tr>
<tr>
<td>24573</td>
<td>0</td>
<td>1 3 4 6 P</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>fw</td>
</tr>
<tr>
<td>24667</td>
<td>0</td>
<td>1 3 4 6 P</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24668</td>
<td>0</td>
<td>1 3 4 6 P</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24670</td>
<td>0</td>
<td>1 3 4 6 P</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24671</td>
<td>0</td>
<td>1 3 4 6 P</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25412</td>
<td>0</td>
<td>1 3 4 6 P</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24642</td>
<td>0</td>
<td>2 3 4 5 6 7 P</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>fwk0 dev</td>
</tr>
<tr>
<td>24643</td>
<td>0</td>
<td>2 3 4 5 6 7 P</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30186</td>
<td>0</td>
<td>all</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>clishd</td>
</tr>
</tbody>
</table>

**Showing Affinity Configurations**

Show the configured affinities of the Virtual System.

Syntax

```
fw ctl affinity -l
```

Example

```
fw ctl affinity -l
```

Output

```
VS_0 cpd: CPU 0 2
VS_0 FWK_INSTANCE_0: CPU 3 7
VS_0 FWK_INSTANCE_4: CPU 3 7
VS_0 fwd: CPU 3 7
VS_0 fwk: CPU 0 1
VS_1: CPU 2 4 6
VS_1 fwd: CPU 4 5
VS_1 fwk: CPU 2 3 4 5 6 7
```
VSX Commands

This section describes the vsx commands.

Note - fw6 vsx commands are not supported.

vsx fetch

Description
Fetches the most current configuration files from the Main Domain Management Server, and applies it to the VSX Gateway.

Syntax
vsx fetch [-v] [-q] [-s] local
vsx fetch [-v | -q| -s] [-f conf_file]
vsx fetch [-v | -q] -C "command"
vsx fetch [-v | -q| -c| -n| -s] [management]

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-c</td>
<td>Cluster mode</td>
</tr>
<tr>
<td>-n</td>
<td>Do not run local.vsall if VSX configuration, as fetched from management server, is up-to-date.</td>
</tr>
<tr>
<td>-s</td>
<td>Concurrent fetches for multi-processor environment.</td>
</tr>
<tr>
<td>-q</td>
<td>Quiet mode - Only summary lines appear.</td>
</tr>
<tr>
<td>-v</td>
<td>Verbose mode - Detailed information appears.</td>
</tr>
<tr>
<td>-f conf_file</td>
<td>Fetches NCS commands configuration file instead of the default local.vsall.</td>
</tr>
</tbody>
</table>

local
Reads local.vsall configuration file from $FWDIR/state/local/vsx and executes the NCS

management
Fetches local.vsall from management, replaces and runs it.

-C command
Execute NCS command

Return Value
0 (zero) indicates that the command executed successfully. Any other response indicates an error.
fw vsx fetch
Fetching VSX Configuration From: 10.18.99.101
Local VSX Configuration is Up-To-Date.
Cleaning un-used Virtual Systems entries (local.vskeep).
Purge operation succeeded.
Fetching Virtual Systems configuration file (local-vsall).
SecureXL device has been enabled for vsid 1
SecureXL device has been enabled for vsid 2
SecureXL device has been enabled for vsid 3
Virtual Systems configuration file installed successfully

### vsx fetchvs

**Description**
Retrieves a specific Virtual System configuration file based on information stored locally on the gateway.

**Syntax**
```
vsx fetchvs [-v | -q] [<vs name> | <vs ID>]
```

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-q</td>
<td>Quiet output. Only summary information appears.</td>
</tr>
<tr>
<td>-v</td>
<td>Verbose output. Detailed information appears.</td>
</tr>
<tr>
<td>&lt;vs name&gt;</td>
<td>Enter the Virtual System name or ID.</td>
</tr>
<tr>
<td>&lt;vs ID&gt;</td>
<td></td>
</tr>
</tbody>
</table>

**Return Value**
0 (zero) indicates that the command executed successfully. Any other response indicates an error.

**Example**
```
fw vsx fetchvs California
```

**Output**
```
fw vsx fetchvs 2
SecureXL device has been enabled for vsid 2
```

### vsx stat

**Description**
Displays VSX status information.

**Syntax**
```
vsx stat [-v] [-l] [<vsid>]
```

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-v</td>
<td>Displays detailed (verbose) information.</td>
</tr>
<tr>
<td>-l</td>
<td>Displays a detailed list of all virtual devices.</td>
</tr>
<tr>
<td>&lt;vsid&gt;</td>
<td>Displays statistics for the specified Virtual System</td>
</tr>
</tbody>
</table>

Check Point VSX Administration Guide R76   |   170
Output

VSX Gateway Status
==================
Name:                 MyGateway
Security Policy:       MyGateway_VSX
Installed at:          10Dec2007 10:31:25
SIC Status:            Trust

Number of Virtual Systems allowed by license:100
Virtual Systems [active/configured]: 2/2
Virtual Routers and Switches [active/configured]:1/1
Total connections [current/limit] 4/46000

virtual devices Status
=======================
ID  | Type & Name | Security Policy | Installed at   | SIC
---+-------------+-----------------+----------------+------
1 | W VSW_1     |<Not Applicable> |                | Trust
2 | S VS1       |Standard         | 10Apr2005 10:31| Trust
3 | S VSs2      |Standard         | 10Apr2005 10:31| Trust

Type: S - Virtual System, B - Virtual System in Bridge mode,
R - Virtual Router, W - Virtual Switch,
? - Information unavailable.

Output for a specific Virtual System

[Expert@sun:51]# vsx stat -l 51

VSID:            51
VRID:            51
Type:            Virtual System
Name:            sun_mem_vs49
Security Policy: TDCH-FW08.OPA
Installed at:    29Jul2009 12:37:58
SIC Status:      Trust
Connections number: 0
Connections peak:  0
Connections limit: 15000

vsx sic reset

Description    Resets SIC for the Virtual System. Run vsenv <vsid> to change context and show an interface list for a different Virtual System.

Syntax         vsx sic reset
Example
csenv 1
csx sic reset

Output
resetting SIC for VSID 1

Note - On the management server, use the `cpca_client revoke_cert` command to cancel the old certificate. In SmartDashboard, open the Virtual System object for editing. Click **OK**. This action creates a new certificate, and transfers the certificate to the gateway.

## Link Aggregation CLI Commands

cphaconf show_bond
Displays the status of an interface bond, or with the `-a` argument, a summary table of all bonds. When a bond is specified information for each slave interface is also displayed.

**Syntax**
cphaconf show_bond {<bond-name>|-a}

**Example**

```
[Expert@GW-1]# cphaconf show_bond bond0
```

<table>
<thead>
<tr>
<th>Bond name:</th>
<th>bond0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bond mode:</td>
<td>Load Sharing</td>
</tr>
<tr>
<td>Bond status:</td>
<td>Up</td>
</tr>
<tr>
<td>Balancing mode:</td>
<td>802.3ad Layer3+4 Load Balancing</td>
</tr>
<tr>
<td>Configured slave interfaces:</td>
<td>4</td>
</tr>
<tr>
<td>In use slave interfaces:</td>
<td>4</td>
</tr>
<tr>
<td>Required slave interfaces:</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Slave Name</th>
<th>Status</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth2</td>
<td>Active</td>
<td>Yes</td>
</tr>
<tr>
<td>eth3</td>
<td>Active</td>
<td>Yes</td>
</tr>
<tr>
<td>eth4</td>
<td>Active</td>
<td>Yes</td>
</tr>
<tr>
<td>eth5</td>
<td>Active</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Report Results**

- Required slave interfaces (“Setting Critical Required Interfaces” on page 132).
- The **Status** column can contain these values:
  - **Down** (Load Sharing mode only) - the physical link is down.
  - **Active** - currently handling traffic.
  - **Standby** (High Availability mode only) - the interface is ready, and can support internal bond failover.
- Not Available (High Availability mode only) - either the physical link is broken, or that the Cluster member is in status *down*. The bond cannot failover in this state.

- The Link column reports whether the physical link exists.

**chpaconf failover_bond**

Initiates bond interface failover in the High Availability mode.

**Syntax**

```
cphaconf failover_bond <bond-name>
```

**cphaprob -a if**

Displays the status of all interface bonds and VLANs for all the Virtual Systems. For a High Availability bond, specifies whether it can failover.

**Syntax**

```
cphaprob -a if
```

**Example**

```
# cphaprob -a if

vsid 0:
--------
Required interfaces: 1
Required secured interfaces: 0

bond1   Inbound: UP  Outbound: DOWN (127 secs)  non sync(non secured), broadcast, bond High Availability
eth0    UP  non sync(non secured), multicast
eth1    DOWN (127 secs)  sync(secured), broadcast

Virtual cluster interfaces: 1

eth0  172.16.6.115

vsid 2:
--------
Required interfaces: 0
Required secured interfaces: 0

eth1    DOWN (38.6 secs)  sync(secured), broadcast
eth3    Inbound: UP  Outbound: DOWN (38.6 secs)  non sync(non secured), multicast

Virtual cluster interfaces: 1

eth3  10.0.0.10
```
**vsx_util Command**

**Description**
Performs various VSX maintenance tasks. You run this command from the expert mode on the management server (Security Management Server or a Main Domain Management Server in a Multi-Domain Security Management environment).

```
vsx_util <sub-command> [parameters]
```

**Syntax**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-s &lt;management IP&gt;</td>
<td>Perform action using the specified management IP</td>
</tr>
<tr>
<td>-u &lt;user name&gt;</td>
<td>Perform the action using the specified administrator</td>
</tr>
<tr>
<td>-c &lt;cluster or gateway name&gt;</td>
<td>Perform the action on the specified cluster or VSX Gateway</td>
</tr>
<tr>
<td>-m &lt;member name&gt;</td>
<td>Perform the action on the specified member</td>
</tr>
<tr>
<td>-h</td>
<td>Display help text</td>
</tr>
</tbody>
</table>

**Comments**
- Note - You must close SmartDashboard before executing the `vsx_util` command if any Virtual Systems are defined on the Security Management Server or Multi-Domain Security Management Domain Management Server. Failure to do so may result in a database locked error.

The `vsx_util` command typically requires you to enter the following information before executing the command:

- Management server name or IP address
- User name and password
- The command may ask for the name of one or more VSX objects upon which the command operates
- Most `vsx_util` sub-commands are interactive and require additional user input. Brief descriptions of additional input requirements appear in the Input section for the various sub-commands. The instructions on the screen typically provide helpful information regarding required information.

**add_member**

**Description**
Adds a new member to an existing VSX cluster.

```
vsx_util add_member
```

**Input**
- VSX cluster object name
- New member name
- IP for [interface]: IP address assigned to specified interface (IP address is required for management and sync network interfaces)
Description: Adds a new member to an existing VSX cluster.

Comments: Run the command and follow the instructions on the screen. When the command finishes executing, you must also Run the vsx_util add_member_reconf command.

See Adding a New Member (on page 94) before using this command.

add_member_reconf

Description: Restores VSX configuration after adding a cluster member

Syntax:

\[ \text{vsx}_\text{util} \ add\_member\_reconf \]

Input:

- VSX member object name: VSX cluster member name
- Activation Key: SIC activation key assigned to the Security Management Server or main Domain Management Server
- Retype Activation Key: Retype to confirm the SIC activation key

Comments: Execute the command and follow the instructions on the screen. Reboot the member after the command script finishes.

Review the procedure for defining a new member (See “Adding a New Member” on page 94) before using this command.

change_interfaces

Description: Automatically replaces designated existing interfaces with new interfaces on all virtual devices to which the existing interfaces connect.

This command is useful when converting a deployment to use Link Aggregation, especially where VLANs connect to many virtual devices.

Syntax:

\[ \text{vsx}_\text{util} \ change\_interfaces \]

Comments:

- This command is interactive. Follow the instructions on the screen.
- This command supports the resume feature.
- You can use this command to migrate a VSX deployment (“Migrating between Servers with Different Interface Names” on page 123) from an Open Server to a Check Point appliance by using the Management Only mode.
- Refer to the notes (“Notes” on page 176) for additional information.

Important - You must close SmartDashboard for all Multi-Domain Security Management Domain Management Servers using the affected interfaces prior to running this command.
**Using vsx_util change_interfaces**

To change interfaces:

2. On the management server, enter the Expert Mode and run the `vsx_util change_interfaces` command.
3. Enter the Security Management Server or Multi-Domain Security Management main Domain Management Server IP address.
4. Enter the administrator name and password as requested.
5. Enter the VSX cluster object name.
6. When prompted, select one of the following options:
   - **Apply changes to management and Security Gateway / cluster members**: Changes the on the management server, the VSX Security Gateway and cluster members.
   - **Apply changes to management Only**: Changes interface on the management server only. You must use the `vsx_util reconfigure` command ("reconfigure" on page 179) to push the updated configuration to VSX Gateways or cluster members.
7. When prompted, select the interface to be replaced.
8. When prompted, select the replacement interface.
   a) You can optionally add a new interface by selecting "Enter new interface name". This interface must physically exist on the VSX Gateway or cluster members or the operation will fail.
   b) At the prompt, enter the new interface name. If the new interface is a bond, the interface name must match the bond name exactly (bond names are case sensitive).
9. To replace additional interfaces, enter "y" when prompted and repeat steps 6 through 8.
10. To complete the process, enter "n".
11. If you selected the **Apply changes to management only** option, run the `vsx_util reconfigure` command ("reconfigure" on page 179) to push the updated configuration to the VSX Gateways or cluster members.
12. Reboot the VSX Gateway and/or cluster members as appropriate.

**Notes**

- The **Apply changes to management and gateway / cluster members** option verifies connectivity between the management server and the VSX Gateway or cluster members. In the event of a connectivity failure one of the following actions occur:
  a) If **all** of the newly changed interfaces fail to establish connectivity, the process terminates unsuccessfully.
  b) If one or more interfaces successfully establish connectivity, while one or more other interfaces fail, you may optionally continue the process. In this case those interfaces for which connectivity was established successfully will be changed. For those interfaces that failed, you must then resolve the issue and then run the `vsx_util reconfigure` ("reconfigure" on page 179) command to complete the process.
- If you select the **Apply changes to management Only** option, you can select another interface from list (if any are available) or select the option to add a new interface.
change_mgmt_ip

**Description**  
Changes gateway or cluster member management IP address

**Syntax**  
```
vsx_util change_mgmt_ip
```

**Input**  
- VSX Gateway / member object name
- New management IP address

**Comments**  
- We recommend that you back up the management database before using this command.
- Execute the command and follow the instructions on the screen.

change_mgmt_private_net

**Description**  
Changes the cluster internal communication network IP address

**Syntax**  
```
vsx_util change_private_net
```

**Input**  
- VSX cluster object name
- New cluster private network: New IP address for the cluster private network

**Comments**  
- We recommend that you back up the management database before using this command.
- The private network IP address must be unique and not used anywhere behind the VSX Gateway, cluster or Virtual Systems.
- The new cluster private network must conform to the net mask 255.255.252.0.
- Execute the command and follow the instructions on the screen.

change_interfaces

**Description**  
Automatically replaces designated existing interfaces with new interfaces on all virtual devices to which the existing interfaces connect.

This command is useful when converting a deployment to use Link Aggregation, especially where VLANs connect to many virtual devices.

**Syntax**  
```
vsx_util change_interfaces
```

**Comments**  
- This command is interactive. Follow the instructions on the screen.
- This command supports the resume feature.
- You can use this command to migrate a VSX deployment ("Migrating between Servers with Different Interface Names" on page 123) from an Open Server to a Check Point appliance by using the Management Only mode.
- Refer to the notes ("Notes" on page 176) for additional information.
**Important** - You must close SmartDashboard for all Multi-Domain Security Management Domain Management Servers using the affected interfaces prior to running this command.

**Using vsx_util change_interfaces**

To change interfaces:

2. On the management server, enter the Expert Mode and run the `vsx_util change_interfaces` command.
3. Enter the Security Management Server or Multi-Domain Security Management main Domain Management Server IP address.
4. Enter the administrator name and password as requested.
5. Enter the VSX cluster object name.
6. When prompted, select one of the following options:
   - **Apply changes to management and Security Gateway / cluster members**: Changes the on the management server, the VSX Security Gateway and cluster members.
   - **Apply changes to management Only**: Changes interface on the management server only. You must use the `vsx_util reconfigure` command ("reconfigure" on page 179) to push the updated configuration to VSX Gateways or cluster members.
7. When prompted, select the interface to be replaced.
8. When prompted, select the replacement interface.
   a) You can optionally add a new interface by selecting "Enter new interface name". This interface must physically exist on the VSX Gateway or cluster members or the operation will fail.
   b) At the prompt, enter the new interface name. If the new interface is a bond, the interface name must match the bond name exactly (bond names are case sensitive).
9. To replace additional interfaces, enter "y" when prompted and repeat steps 6 through 8.
10. To complete the process, enter "n".
11. If you selected the **Apply changes to management only** option, run the `vsx_util reconfigure` command ("reconfigure" on page 179) to push the updated configuration to the VSX Gateways or cluster members.
12. Reboot the VSX Gateway and/or cluster members as appropriate.

**Notes**

- The **Apply changes to management and gateway / cluster members** option verifies connectivity between the management server and the VSX Gateway or cluster members. In the event of a connectivity failure one of the following actions occur:
  a) If all of the newly changed interfaces fail to establish connectivity, the process terminates unsuccessfully.
  b) If one or more interfaces successfully establish connectivity, while one or more other interfaces fail, you may optionally continue the process. In this case those interfaces for which connectivity was established successfully will be changed. For those interfaces that
failed, you must then resolve the issue and then run the vsx_util reconfigure ["reconfigure" on page 179] command to complete the process.

- If you select the Apply changes to management Only option, you can select another interface from list (if any are available) or select the option to add a new interface.

**change_mgmt_subnet**

**Description**  
Change the gateway or member management subnet

**Syntax**  
```
vsx_util change_mgmt_subnet
```

**Input**  
- VSX Gateway / member object name
- New subnet mask

**Comments**  
- Backup the management database before using this command
- Only automatically generated routes are changed by the command script. You must remove and/or change all manually created routes using the previous management subnet.

To perform this action, execute the command and follow the instructions on the screen. Reboot the VSX Gateway or cluster members after the command script finishes.

**convert_cluster**

**Description**  
Converts the cluster type from High Availability to VSLS or from VSLS to High Availability

**Syntax**  
```
vsx_util convert_cluster
```

**Input**  
VSX cluster object name

ClusterXL mode: **HA** for High Availability or **LS** for Virtual System Load Sharing

**Comments**  
Backup the management database before using this command.

To perform this action, execute the command and follow the instructions on the screen.

When switching to High Availability, all Virtual Systems are active on the same member by default. Peer Virtual Systems are standby on other members.

When converting to VSLS, all members must be in the Per Virtual System state.

**reconfigure**

**Description**  
Restores a VSX configuration to a newly installed gateway or cluster member

**Syntax**  
```
vsx_util reconfigure
```
Input

VSX cluster member name  
SIC activation key assigned to the Security Management Server or Domain Management Server  
Retype to confirm the SIC activation key

Comments

For more about how to use the vsx_util reconfigure command, go to sk97552 http://supportcontent.checkpoint.com/solutions?id=sk97552.

This command is also useful for restoring a gateway or cluster member after a system failure.

Execute the command and follow the instructions on the screen.

A new gateway or cluster member must have the same hardware specifications and configuration as its replacement and other cluster members. Most importantly, it must have the same number of interfaces (or more) and the same management IP address.

The new or replacement machine must be a new installation. You cannot use a machine with a previous VSX configuration.

remove_member

Description  
Removes a member from an existing cluster

Syntax

vsx_util remove_member

Comments  
Backup the management database before using this command  
Make certain that you remove member license before executing this command  
Execute the command and follow the instructions on the screen

show_interfaces

Description  
Displays selected interface information in a VSX deployment. Provides information regarding interface types, connections to virtual devices, and IP addresses. The output appears on the screen and is also saved to the interfacesconfig.csv file.

Syntax

vsx_util show_interfaces

Parameters

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) All Interfaces</td>
<td>Show all interfaces (physical and Warp)</td>
</tr>
<tr>
<td>2) All Physical Interfaces</td>
<td>Show Physical interfaces only</td>
</tr>
<tr>
<td>3) All Warp Interfaces</td>
<td>Show Warp interfaces only</td>
</tr>
<tr>
<td>4) A Specific Interface</td>
<td>Enter the interface name when prompted to a specific interface.</td>
</tr>
</tbody>
</table>
**Note** - You cannot specify a VLAN tag as a parameter for the **Specific Interface** option. You can, however, specify an interface used as a VLAN (without the tag suffix) to view all tags associated with that interface. This is illustrated in the sample output below.

### Sample Output

Which interface would you like to display?
1) All Interfaces
2) All Physical Interfaces
3) All Warp Interfaces
4) A Specific Interface

Enter your choice: 4

Enter Interface Name: eth1

<table>
<thead>
<tr>
<th>Type &amp; Interface</th>
<th>virtual device Name</th>
<th>VSID</th>
<th>IP Address</th>
<th>Netmask</th>
</tr>
</thead>
<tbody>
<tr>
<td>V eth1.11</td>
<td>vs1</td>
<td>2</td>
<td>10.1.1.11</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>V eth1.44</td>
<td>vs4</td>
<td>6</td>
<td>10.4.4.44</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>V eth1.22</td>
<td>vs2</td>
<td>3</td>
<td>10.2.2.22</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>V eth1.33</td>
<td>vs3</td>
<td>4</td>
<td>10.3.3.33</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>V eth1.55</td>
<td>vs5</td>
<td>5</td>
<td>10.5.5.55</td>
<td>255.255.255.0</td>
</tr>
</tbody>
</table>

Type: M - Management Interface  S - Synchronization Interface
V - VLAN Interface             W - Warp Interface
U - Used Interface             A - Available Interface
X - Unknown Interface          E - Error in Interface Properties

### Upgrade

**Description**
Upgrades Gateways and/or cluster members to newer versions

**Syntax**
```
svx_util upgrade
```

**Comments**
This command updates all VSX objects in the management database to the designated newer version.

Backs up the management server.

Execute the command and follow the instructions on the screen.

After the command script finishes, execute the vsx_util reconfigure command.

### view_vs_conf

**Description**
Displays virtual device configuration and status, including troubleshooting information. This command also compares the management server database with the actual VSX Gateways and cluster member configurations.

**Syntax**
```
svx_util view_vs_conf
```

**Output**
### Interfaces configuration table:

<table>
<thead>
<tr>
<th>Interfaces</th>
<th>Mgmt</th>
<th>VSX GW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Address</td>
<td>Mask</td>
</tr>
<tr>
<td>eth1.51</td>
<td>20.10.10.1</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>wrp128</td>
<td>172.23.50.181</td>
<td>255.255.255.0</td>
</tr>
</tbody>
</table>

**Interfaces Table Legend:**

- V: Interface exists on the gateway and matches management information.
- N/A: Fetching virtual device configuration from the gateway failed.
- !IP: Interface exists on the gateway, but there is an IP address mismatch.
- !MASK: Interface exists on the gateway, but there is a net mask mismatch.

### Routing table:

<table>
<thead>
<tr>
<th>Routes</th>
<th>Mgmt</th>
<th>VSX GW(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination</td>
<td>Mask</td>
<td>Gateway</td>
</tr>
<tr>
<td>172.23.50.0</td>
<td>255.255.255.0</td>
<td>wrp128</td>
</tr>
<tr>
<td>20.10.10.0</td>
<td>255.255.255.0</td>
<td>eth1.51</td>
</tr>
<tr>
<td>20.30.30.0</td>
<td>255.255.255.0</td>
<td>172.23.50.82</td>
</tr>
</tbody>
</table>

**Routing Table Legend:**

- V: Route exists on the gateway and matches management information.
- N/A: Fetching virtual device configuration from the gateway failed.
- !NH: Route exists on the gateway, but there is a Next Hop mismatch.
Description Display VSLS load sharing configuration and status
Usage vsx_util vsls
Output
Sample output:
vsx_util vsls
Enter SmartCenter Server/main Domain Management Server IP address (Hit 'ENTER' for 'localhost'):
Enter Administrator Name:
Enter Administrator Password:
Enter ClusterXL Load Sharing cluster object name:

<table>
<thead>
<tr>
<th>VSID</th>
<th>VS name</th>
<th>m5</th>
<th>m6</th>
<th>m7</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>vs1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>vs2</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>12</td>
<td>vs3</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>13</td>
<td>vs4</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>15</td>
<td>vs5</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>20</td>
<td>vs6</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>10</td>
</tr>
</tbody>
</table>

----+---------+-----------+-----------+-----------+--------+
Total weight | 20        | 20        | 20        | 60     |
----+---------+-----------+-----------+-----------+--------+
Legend:
0  - Highest priority
1  - Next priority
2  - Lowest priority

vsls

Description Displays the Virtual System Load Sharing Menu, which allows you to perform a variety of configuration tasks for Load Sharing deployments. You perform configuration tasks ("Configuring Virtual System Load Sharing" on page 99) interactively by following the instructions on the screen.

Syntax
vsx_util vsls

Output
VS Load Sharing - Menu
1. Display current VS Load sharing configuration
2. Distribute all Virtual Systems so that each cluster member is equally loaded
3. Set all VSs active on one member
4. Manually set priority and weight
5. Import configuration from a file
6. Export configuration to a file
7. Exit

Enter redistribution option (1-7) [1]

Comments
- This command is interactive. Select the desired menu option and follow the instructions on the screen.

You use the `vsx_util vsls` command to perform various Virtual System Load Sharing configuration tasks, including:

1. Displaying the current VSLS configuration
2. Distributing Virtual Systems equally amongst cluster members
3. Set all Virtual Systems as active on one member
4. Manually define the priority and weight for individual Virtual Systems
5. Import VSLS configurations from comma separated value (CSV) text files
To work with the \texttt{vsx\_util vsls} command:

1. Run \texttt{vsx\_util vsls} from the Expert mode on the management server
2. Select the desired choice from the \textbf{VSLs} menu

### The \texttt{cphaprob} Command

You use the \texttt{cphaprob} command to verify cluster functionality and to debug cluster related problems. This section provides a brief overview of the \texttt{cphaprob} command and its command options. For complete documentation and use cases, refer to the \textit{R76 ClusterXL Administration Guide} [http://supportcontent.checkpoint.com/documentation_download?ID=22910].

A critical device is a process running on a cluster member that enables the member to notify other cluster members that it can no longer function as a member. The device reports to the ClusterXL mechanism regarding its current state or it may fail to report, in which case ClusterXL decides that a failure has occurred and another cluster member takes over. When a critical device (also known as a Problem Notification, or pnote) fails, the cluster member is considered to have failed.

There are a number of built-in critical devices, and the administrator can define additional critical devices. The default critical devices are:

\textbf{Cluster interfaces on the cluster members.}

\textbf{Synchronization} — full synchronization completed successfully.

\textbf{Filter} — the Security Policy, and whether it is loaded.

\textbf{fwd} — the VPN-1 daemon.

You can include these commands in scripts for automatic execution.

To produce a usage printout for \texttt{cphaprob} that shows all the available commands, type \texttt{cphaprob} at the command line and press \texttt{Enter}. The following output appears:

```
cphaprob state
\texttt{cphaprob [-a] [-vs vsid] if}

The following commands are NOT applicable for 3rd party:
cphaprob -d <device> -t <timeout(sec)> -s <ok|init|problem> [-p] register
cphaprob -f <file> register
cphaprob -d <device> [-p] unregister
cphaprob -a unregister
cphaprob -d <device> -s <ok|init|problem> report
cphaprob [-i[a]] [-e] [-vs vsid] list
cphaprob -vs <vsid> register
cphaprob -vs <vsid> unregister
cphaprob igmp .................. IGMP membership status
cphaprob [-reset] [-a] ldstat ... Sync serialization statistics
cphaprob [-reset] [-a] synccstat . Sync transport layer statistics
cphaprob fcusstat ................ Full connectivity upgrade statistics
cphaprob tablestat .............. Cluster tables
```

### Command Options

The following table describes the available command options:
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[-vs vsid] stat</td>
<td>View the status all cluster members or for a specific Virtual System. -vs is relevant only for VSLS.</td>
</tr>
<tr>
<td>[-a] [-vs vsid] if</td>
<td>View the state of the cluster member interfaces and the virtual cluster interfaces. -vs is relevant only for VSLS.</td>
</tr>
<tr>
<td>[-i[a]] [-e] list</td>
<td>View the list of critical devices on a cluster member, and of all the other machines in the cluster.</td>
</tr>
<tr>
<td>-d &lt;device&gt; -t &lt;timeout (sec) &gt; -s &lt;ok</td>
<td>init</td>
</tr>
<tr>
<td>-f &lt;file&gt; register</td>
<td>Register all the user defined critical devices listed in &lt;file&gt;.</td>
</tr>
<tr>
<td>-d &lt;device&gt; [-p] unregister</td>
<td>Unregister a user defined &lt;device&gt; as a critical process. This means that this device is no longer considered critical.</td>
</tr>
<tr>
<td>-a unregister</td>
<td>Unregister all user defined devices</td>
</tr>
<tr>
<td>-d &lt;device&gt; -s &lt;ok</td>
<td>init</td>
</tr>
<tr>
<td>[-reset] ldstat -vs</td>
<td>View sync serialization statistics. -vs is relevant only for VSLS.</td>
</tr>
<tr>
<td>[-reset] syncstat -vs</td>
<td>View sync transport layer statistics. -vs is relevant only for VSLS.</td>
</tr>
<tr>
<td>tablestat</td>
<td>Displays interfaces and IP addresses for each cluster member</td>
</tr>
</tbody>
</table>