



Check Point
SOFTWARE TECHNOLOGIES LTD.

16 May 2017

CHECK POINT VSX

R80.10

Administration Guide

Classification: [Protected]



Check Point
SOFTWARE TECHNOLOGIES LTD.

INFINITY

© 2017 Check Point Software Technologies Ltd.

All rights reserved. This product and related documentation are protected by copyright and distributed under licensing restricting their use, copying, distribution, and decompilation. No part of this product or related documentation may be reproduced in any form or by any means without prior written authorization of Check Point. While every precaution has been taken in the preparation of this book, Check Point assumes no responsibility for errors or omissions. This publication and features described herein are subject to change without notice.

RESTRICTED RIGHTS LEGEND:

Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 and FAR 52.227-19.

TRADEMARKS:

Refer to the Copyright page <http://www.checkpoint.com/copyright.html> for a list of our trademarks.

Refer to the Third Party copyright notices http://www.checkpoint.com/3rd_party_copyright.html for a list of relevant copyrights and third-party licenses.

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.



Check Point R80.10

For more about this release, see the R80.10 home page
<http://supportcontent.checkpoint.com/solutions?id=sk111841>.



Latest Version of this Document

Download the latest version of this document
<http://downloads.checkpoint.com/dc/download.htm?ID=54764>.

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



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 R80.10 Administration Guide.



Searching in Multiple PDFs

To search for text in all the R80.10 PDF documents, download and extract the complete R80.10 documentation package

<http://downloads.checkpoint.com/dc/download.htm?ID=54846>.

Use **Shift-Control-F** in Adobe Reader or Foxit reader.

Revision History

Date	Description
16 May 2017	First release of this document

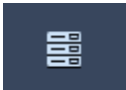
SmartConsole Toolbars

For a guided tour of SmartConsole, click **What's New** in the left bottom corner of SmartConsole.

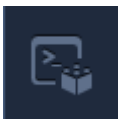
Global Toolbar (top left of SmartConsole)

	Description and Keyboard Shortcut
	The main SmartConsole Menu
	The Objects menu. Also leads to the Object Explorer Ctrl+E
	Install policy on managed gateways Ctrl+Shift+Enter

Navigation Toolbar (left side of SmartConsole)

	Description and Keyboard Shortcut
	Gateways & Servers configuration view Ctrl+1
	Security Policies Access Control view Security Policies Threat Prevention view Ctrl+2
	Logs & Monitor view Ctrl+3
	Manage & Settings view - review and configure the Security Management Server settings Ctrl+4

Command Line Interface Button (left bottom corner of SmartConsole)

	Description and Keyboard Shortcut
	Open a command line interface for management scripting and API F9

What's New Button (left bottom corner of SmartConsole)

	Description and Keyboard Shortcut
	Open a tour of the SmartConsole

Objects and Validations Tabs (right side of SmartConsole)

	Description
Objects	Manage security and network objects
Validations	Validation warnings and errors

System Information Area (bottom of SmartConsole)

	Description
Task List	Management activities, such as policy installation tasks
Server Details	The IP address of the Security Management Server
Connected Users	The administrators that are connected to the Security Management Server

Contents

Important Information.....	3
SmartConsole Toolbars	4
Terms	13
Introduction to VSX.....	15
VSX Overview.....	15
How VSX Works	15
Physical Network Topology	16
VSX Virtual Network Topology	17
VSX Architecture and Concepts.....	18
The VSX Gateway.....	18
Management Server Connections.....	18
Management Interface	22
Virtual Devices	23
Virtual System	23
Virtual Routers	23
Virtual Switches.....	24
Interfaces	24
Physical Interfaces	26
VLAN Interfaces.....	26
Warp Links	26
Unnumbered Interfaces.....	27
VSX Management Overview.....	28
Security Management Server Model.....	28
Multi-Domain Security Management Model.....	28
Management Model Comparison	30
Management Server Communication - SIC	30
VSX Traffic Flow	30
Overview.....	30
Context Determination	31
Security Enforcement.....	34
Forwarding to Destination	34
VSX Routing Concepts	34
Routing Overview.....	34
Routing Between Virtual Systems	34
Source-Based Routing.....	38
NAT.....	38
Dynamic Routing.....	38
VSX Clusters.....	39
High Availability.....	39
Virtual System Load Sharing (VSLS)	39
Deploying VSX	40
Internal Network Deployment Strategies	40
Security Gateway Deployment on a Physical Network	40
VSX Virtual System Deployment Strategies	41
Physical Internal Interface for Each Virtual System	41
Virtual Systems with Internal VLAN Interfaces.....	42
Internal Virtual Router with Source-Based Routing	43

Virtual Systems in Bridge Mode.....	44
Active/Standby Bridge Mode.....	44
Organizational Deployment Strategies	47
Enterprise Deployments.....	47
Managed Service Providers Using Multi-Domain Security Management	51
Data Centers.....	53
Migrating between Servers with Different Interface Names	56
Configuring VSX	58
Overview.....	58
Rules & Security Policies	59
Configuring VSX Gateways	59
Configuring 64-Bit Virtual System Support.....	59
Working with VSX Gateways.....	60
Changing VSX Gateway Definitions	60
Deleting a VSX Gateway	63
VSX Gateway Recovery	63
Working with Virtual Systems.....	63
Creating a New Virtual System	65
Modifying a Virtual System	67
Deleting a Virtual System	69
Working with Virtual Switches	69
Modifying a Virtual Switch	70
Creating a New Virtual Switch.....	71
Working with Virtual Routers.....	71
Creating a New Virtual Router	74
Modifying a Virtual Router Definition	74
Deleting a Virtual Router	75
Working with Source-Based Routing	75
CoreXL for Virtual Systems.....	77
Configuring CoreXL on a VSX Gateway.....	78
Configuring CoreXL on Virtual Systems.....	78
Dynamic Routing for Virtual Devices.....	79
Working with Interface Definitions.....	79
Adding a New Interface	79
Changing an Interface Definition	82
Deleting an Interface	82
Working with Authentication.....	82
Supported Authentication Schemes.....	82
Configuring SecurID ACE/Server.....	83
Configuring RADIUS or TACACS/TACACS+	87
Tracking Activity.....	89
Working with Network Address Translation	89
Configuring NAT	89
Using Application Control and URL Filtering with VSX.....	90
Using Anti-Bot and Anti-Virus with VSX	90
Licensing VSX.....	91
VSX Gateway/Cluster Member Licenses.....	91
Upgrading Licenses.....	91
The Trial Period.....	92
Virtual System in Bridge Mode.....	92
Core Network Security	92

Three Layer Hierarchical Model	93
Configuring Virtual Systems for Active/Standby Bridge Mode	93
Enabling Active/Standby Bridge Mode for a New Member	94
Enabling Active/Standby Bridge Mode for Existing Members	94
Enabling Active/Active Bridge Mode when Creating Member	94
Enabling Active/Active Bridge Mode for Existing Members	94
Custom Configuration or Override in Bridge Mode	94
VLAN Shared Interface Deployment	95
VSX Clusters	96
Separate Interfaces in Bridge Mode	96
Virtual System Load Sharing (VSLS)	97
Using VSX with Multi-Domain Security Management	100
Overview	100
VSX Provisioning	101
Defining Multi-Domain Servers	101
Working with Virtual Devices	102
Adding a Virtual System to a Domain Management Server	102
Adding Virtual Routers and Virtual Switches to a Domain Management Server	102
Introduction to VSX Clusters	103
VSX Clustering Overview	103
Physical Clusters	103
VSX Clusters	104
Planning a Cluster Deployment	105
VSX Cluster Architecture	105
VSX High Availability	106
VSX Gateway High Availability	106
Virtual System Load Sharing (VSLS)	107
Requirements	107
Conceptual Overview	107
Failure Recovery	113
Bridge Mode	113
Spanning Tree Protocol (STP) Bridge Mode	113
Active/Standby Bridge Mode	113
Using Virtual Switches in a Cluster	115
Working with VSX Clusters	117
Configuration Overview	117
Creating VSX Clusters	117
Creating a New Cluster	117
Modifying a Cluster Definition	120
General Properties	120
Cluster Members	120
ClusterXL	121
Creation Templates	122
Physical Interfaces	122
Synchronization	122
Topology	122
NAT	124
VSX Bridge Configuration	125
Changing the Cluster Management IP and/or Subnet	125
Changing the Internal Communication Network IP	125
Working with Cluster Members	125
Adding a New Member	125

Deleting a Member	126
Upgrading Cluster Members	127
Changing the Cluster Type	129
Converting from VSLS to High Availability	129
Converting from High Availability to VSLS	130
Sample Command Output	131
Enabling VSX Gateway High Availability	131
Configuring New Cluster Members	131
Configuring Virtual System Load Sharing	132
Enabling VSLS	132
Creating a New VSLS Cluster	132
Using the vsx_util vsls Command	133
Distributing Virtual Systems Amongst Members	134
Viewing VSLS Status	135
Exporting and Importing VSLS Configurations	136
Configuring Virtual Systems in Bridge Mode	138
Overview	139
STP Bridge Mode	139
Active/Standby Bridge Mode	140
Multi Bridges	141
Advanced Clustering Configuration	143
Clusters on the Same Layer-2 Segment	143
Monitoring all VLANs with ClusterXL	144
Enabling Broadcast Mode	144
Configuring CoreXL in a VSX VSLS Cluster	145
Working with Link Aggregation	146
Link Aggregation Overview	146
Link Aggregation Terminology	146
How Link Aggregation Works	147
High Availability Overview	147
Load Sharing Overview	148
Bond Failover	149
Failover Support for VLANs	149
Bond Interface & Interface Limitations	150
Configuring High Availability Mode	150
Configuring the High Availability Bond	151
Updating the Interface Topology	151
Configuring Load Sharing Mode	153
Configuring the Load Sharing Bond	153
Setting Critical Required Interfaces	154
Setting Affinities	154
Configuring Cisco Switches for Load Sharing	155
For 802.3ad:	155
For XOR:	156
Troubleshooting Bonded Interfaces	156
Troubleshooting Workflow	156
Gaia CLI Reference	157
Link Aggregation - CLI (bonding)	157
Creating or Deleting a Bond Interface	158
Defining the Bond Operating Mode	158
Defining Interfaces	159
Defining the Primary Slave Interface	159

Defining the Media Monitoring Interval.....	160
Defining the UP and Down Delay Times	160
Defining Load Sharing Parameters	160
Making Sure that Link Aggregation is Working.....	161
Optimizing VSX	162
QoS Enforcement	162
Differentiated Services Support	163
Inbound Prioritization.....	163
Policy with Global Scope.....	163
Resource Allocation	163
Latency.....	164
WRED	164
QoS Management.....	164
QoS Configuration.....	165
Monitoring Memory Resources	170
Managing fw vsx mstat	170
Memory Resources for Each virtual device.....	171
Configuring Swap-in Sample Rate.....	172
Using Debug Mode.....	173
VSX CPU Monitoring Commands	173
fw vsx resctrl load_configuration	173
fw vsx resctrl monitor	174
fw vsx resctrl reset.....	174
fw vsx resctrl stat.....	175
Sample Output fw vsx resctrl -u	176
SNMP Monitoring	176
VSX SNMP Modes	176
Supported SNMP Versions.....	177
Enabling VS Mode	177
Sample SNMP VS Mode Configuration	177
SNMP Monitoring of VSX	178
VSX SNMP Modes	178
Supported SNMP Versions.....	181
Configuring SNMP VSX modes.....	181
The VSX SNMP Tree.....	184
Configuring Jumbo Frames	185
Jumbo Frames on a Virtual System.....	185
Jumbo Frames on a Virtual Switch	186
Jumbo Frames on a Virtual Router.....	186
Jumbo Frames on a Virtual System in Bridge Mode	186
VSX Diagnostics and Troubleshooting.....	187
General Troubleshooting Steps	187
Troubleshooting Specific Problems	189
Cannot Establish SIC Trust for Gateway or Cluster	189
SIC Trust Problems with New virtual devices.....	189
Re-establishing SIC Trust with virtual devices	190
Install Policy Error Using VSX Creation Wizard	190
Internal Host Cannot Ping Virtual System	191
Command Line Reference	192
vsx_provisioning_tool.....	192
Transactions.....	193
vsx_provisioning_tool Commands.....	194

Script Examples	201
VSX Gaia CLI Commands	202
Changing the Context	202
Configuring Roles on virtual devices	202
Showing virtual devices	204
CoreXL Commands (fw ctl affinity)	205
Affinity Exceptions	205
Affinity Priorities	205
Setting Affinities	205
Affinity Monitoring	207
Firewall Commands	209
fw getifs	209
fw monitor	210
fw tab	210
fw fetch	212
VSX Commands	213
vsx fetch	213
vsx fetchvs	214
vsx stat	214
vsx sic reset	216
Link Aggregation CLI Commands	216
cphaconf show_bond	216
chpaconf failover_bond	217
cphaprob -a if	217
vsx_util Command	218
add_member	219
add_member_reconf	220
change_interfaces	220
change_mgmt_ip	221
change_interfaces	222
change_mgmt_subnet	223
change_private_net	224
convert_cluster	224
reconfigure	224
remove_member	225
show_interfaces	225
Upgrade	226
view_vs_conf	227
vsls	228
The cphaprob Command	229
Command Options	230

Terms

Active/Standby

A High Availability cluster where only one member handles connections.

Administrator

A SmartConsole user with permissions to manage Check Point security products and the network environment.

Bond

A virtual interface that contains (enslaves) two or more physical interfaces for redundancy and load sharing. The physical interfaces share one IP address and one MAC address.

Bridge Mode

A Security Gateway or Virtual System that works as a Layer-2 bridge device for easy deployment in an existing topology.

Cluster

1. Two or more Security Gateways or servers synchronized for High Availability or Load Sharing.
2. In a virtualized environment - a set of ESX/i hosts used for High Availability or Load Sharing.

Cluster Member

A Security Gateway that is part of a cluster.

ClusterXL

Check Point software-based cluster solution for Security Gateway redundancy and Load Sharing.

Link Aggregation

A technology that joins multiple physical interfaces together into one virtual interface, known as a bond interface. Also known as Interface Bonding.

Management Server

A Security Management Server or a Multi-Domain Server.

Multi Domain Log Server

Physical server that hosts all Domain log servers.

Multi-Domain Security Management

A centralized management solution for large-scale, distributed environments with many different Domain networks.

Multi-Domain Server

A physical server that hosts all Domain servers.

Permission Profile

A predefined group of SmartConsole access permissions assigned to Domains and administrators. With this feature you can configure complex permissions for many administrators with one definition.

Security Gateway

A computer or an appliance that inspects traffic and enforces Security Policies for connected network resources.

Security Management Server

The server that manages, creates, stores, and distributes the security policy to Security Gateways.

SmartDashboard

A legacy Check Point client used to create and manage the security policy.

Standby Multi-Domain Server

All Multi-Domain Servers in a High Availability deployment that cannot manage global policies and objects. Standby Multi-Domain Servers are synchronized with the active Multi-Domain Server.

Traffic

The flow of data between network resources.

Virtual Device

A logical object that emulates the functionality of a type of physical network object.

Virtual Router

A virtual device that functions as a physical router.

Virtual Switch

Also vSwitch. A software abstraction of a physical Ethernet switch. It can connect to physical switches through physical network adapters to join virtual networks with physical networks. It can also be a Distributed Virtual Switch (dvSwitch), for definition and use on multiple ESXi hosts.

Virtual System

A virtual device that implements the functionality of a Security Gateway.

VLAN

Virtual Local Area Network. Open servers or appliances connected to a virtual network which are not physically connected to the same network.

VLAN Trunk

A connection between two switches that contains multiple VLANs.

VPN

Virtual Private Network. A secure, encrypted connection between networks and remote clients on a public infrastructure, to give authenticated remote users and sites secured access to an organization's network and resources.

VSLS

Virtual System Load Sharing. A VSX cluster technology that assigns Virtual System traffic to different active cluster members.

VSX

Virtual System Extension. Check Point virtual networking solution, hosted on a computer or cluster with virtual abstractions of Check Point Security Gateways and other network devices. These virtual devices provide the same functionality as their physical counterparts.

VSX Gateway

Physical server that hosts VSX virtual networks, including all **virtual devices** that provide the functionality of physical network devices. It holds at least one Virtual System, which is called VS0.

Warp Link

An interface between a Virtual System and a Virtual Switch or Virtual Router that is created automatically in a VSX topology.

Introduction to VSX

In This Section:

VSX Overview.....	15
How VSX Works.....	15

VSX Overview

VSX (Virtual System Extension) is a security and VPN solution for large-scale environments. 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 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 for enterprises and service providers. The management server can be installed on a different machine than VSX, or on the same machine.

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.

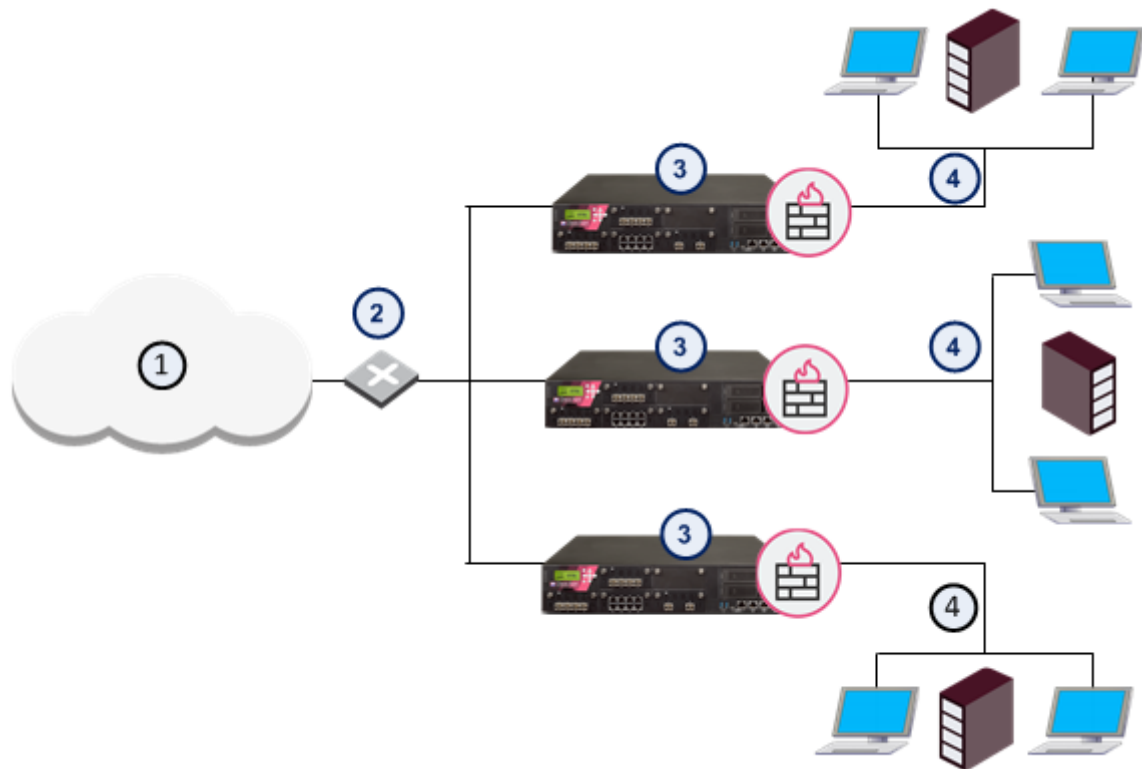
How VSX Works

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

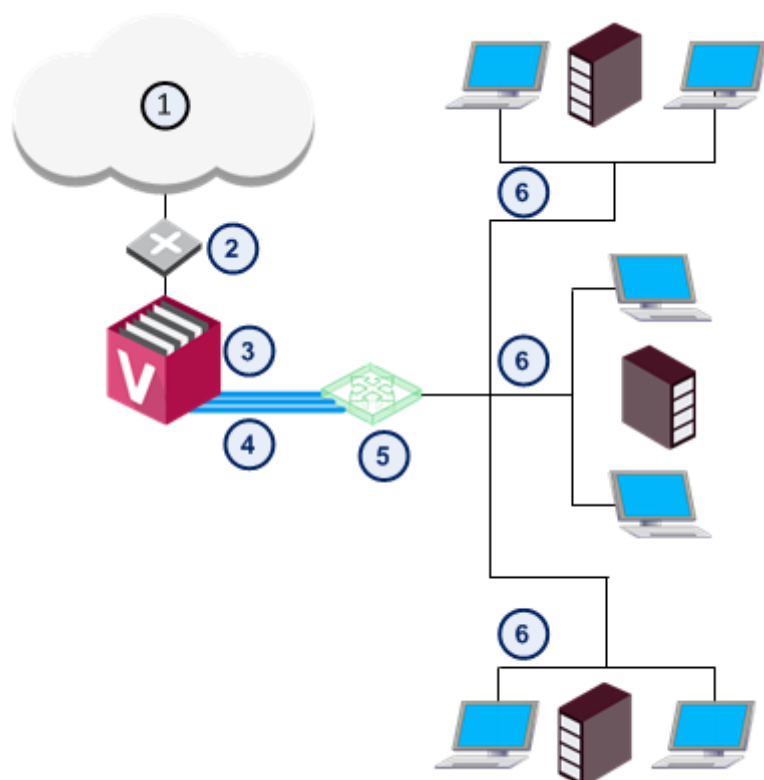
In a typical deployment with multiple Security Gateways, each protects a separate network. Each physical Security Gateway has interfaces to the perimeter router and to the network it protects.



Item	Description
1	Internet
2	Router
3	Security Gateways
4	Network

VSX Virtual Network Topology

Deploy one VSX Gateway with four Virtual Systems to protect multiple networks.



Item	Description
1	Internet
2	Router
3	VSX Gateway. Each Virtual System in a VSX environment is a Security Gateway, with the same security and networking functionality as a physical gateway. Each handles packet traffic to and from the one network it protects.
4	Warp Links. Virtual interfaces and network cables connect the Virtual Systems and the Virtual Switch.
5	Virtual Switch. Connects all the Virtual Systems to the Internet router.
6	Networks

VSX Architecture and Concepts

In This Section:

The VSX Gateway.....	18
Virtual Devices	23
Interfaces	24
VSX Management Overview.....	28
VSX Traffic Flow	30
VSX Routing Concepts	34
VSX Clusters.....	39

The VSX Gateway

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

A VSX Gateway handles these tasks:

- Communicates with the management server to deploy, configure, and manage all virtual devices.
- Manages state synchronization for High Availability and for Load Sharing in cluster deployments.

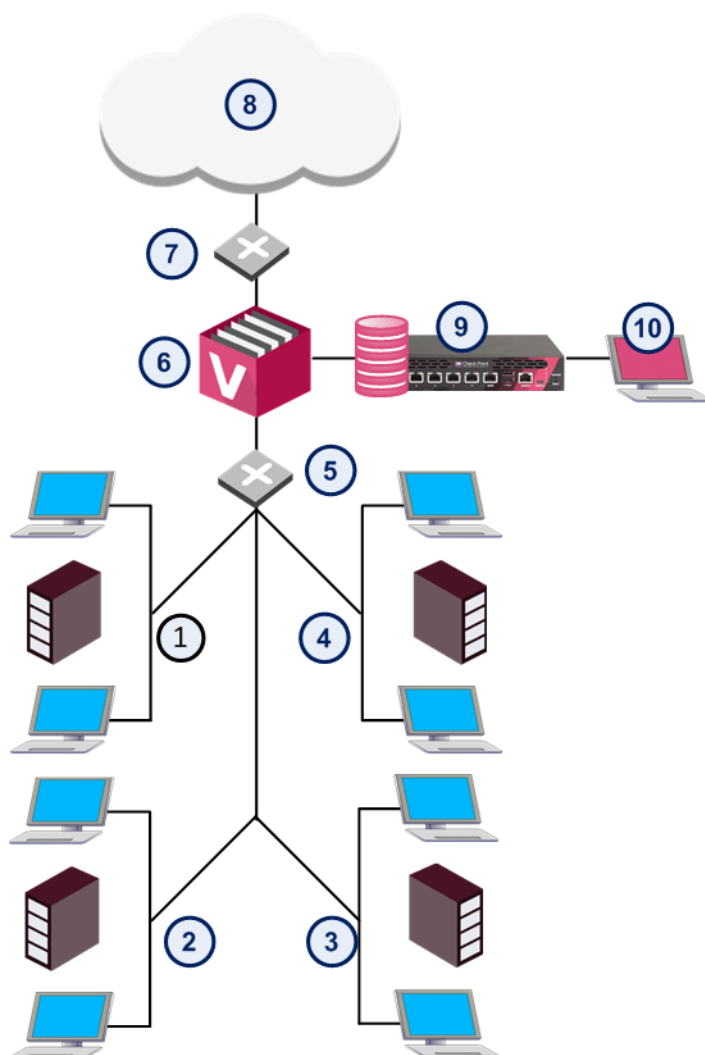
Management Server Connections

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.

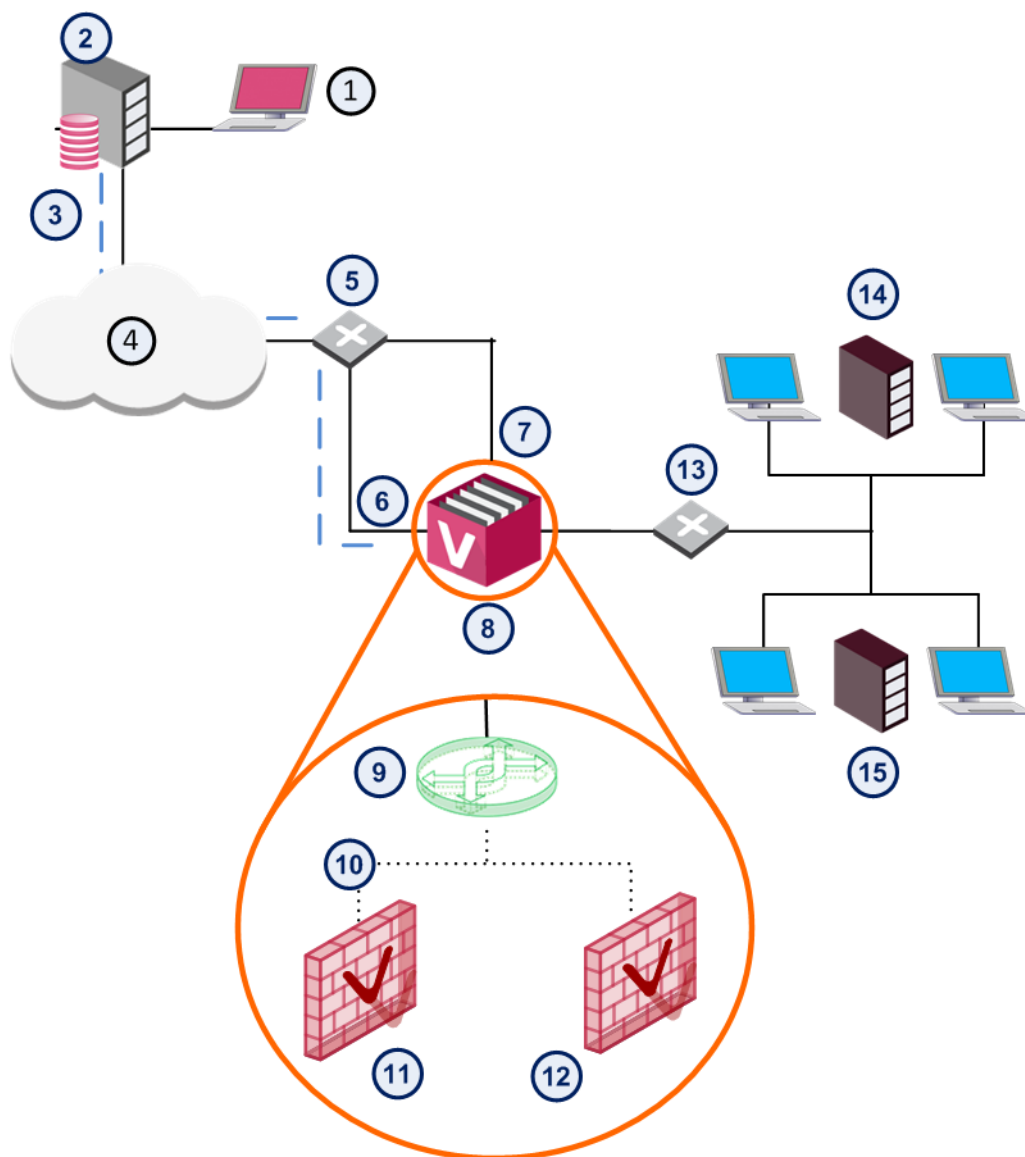


Item	Description
1	Network 1
2	Network 2
3	Network 3
4	Network 4
5	Switch
6	VSX Gateway
7	Router
8	Internet
9	Security Management Server
10	SmartConsole

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.



Item	Description
1	SmartConsole
2	Security Management Server
3	Management traffic
4	Internet
5	Router
6	Dedicated management interface (eth0)
7	External interface
8	VSX Gateway

Item	Description
9	Virtual Switch
10	Warp Link
11	Virtual System 1
12	Virtual System 2
13	Switch
14	Network 1
15	Network 2

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 independently. Each Virtual System maintains its own Software Blades, interfaces, IP addresses, routing table, ARP table, and dynamic routing configuration. Each Virtual System also maintains its own:

- **State Tables:** Each Virtual System has its own kernel tables with configuration and runtime data, such as active connections and IPSec tunnel information.
- **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 and on the VSX Gateway.
- **Configuration Parameters:** Each Virtual System maintains its own configuration, such as IPS settings and TCP/UDP time-outs. 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 runs logging according to its own rules and configuration.

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.



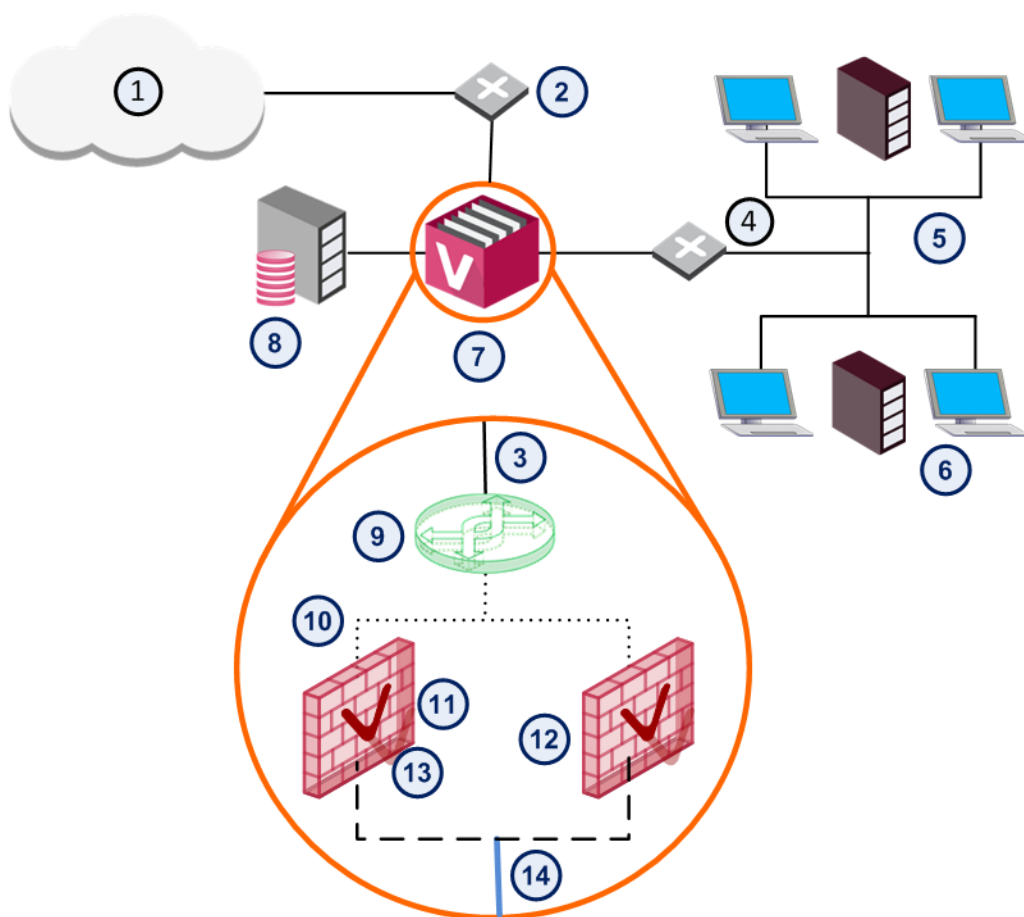
Note - When sharing a physical interface via a Virtual Switch, the IP addresses for Virtual Systems connected to a Virtual Switch should be allocated from the same subnet as the shared interface.

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)



Item	Description
1	Internet
2	Router
3	Physical interface
4	VLAN Switch
5	Network 1
6	Network 2
7	VSX Gateway

Item	Description
8	Security Management Server
9	Virtual Switch
10	Warp interface
11	Virtual System 1
12	Virtual System 2
13	VLAN Interface
14	VLAN Trunk

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 management server and to internal and external networks. There are different types of physical interfaces 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, the management connection arrives through 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.

You can install and configure more physical interfaces to a 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

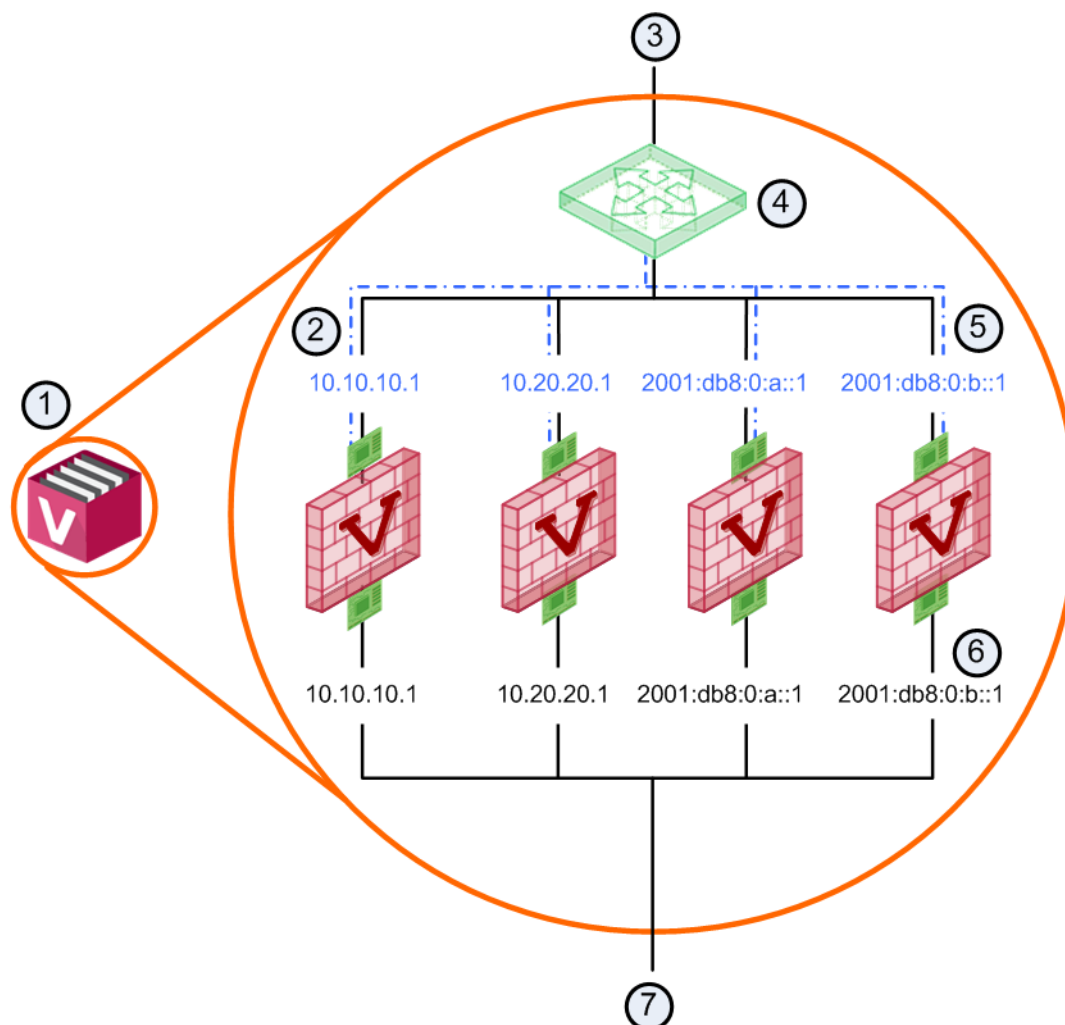
A Warp Link is a virtual point-to-point connection between a Virtual System and a Virtual Router or Virtual Switch. Each side of a Warp Link represents a virtual interface with the appropriate virtual device.

R80.10 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 `wrp` and those on the Virtual Router / Virtual Switch side are assigned the prefix `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 **Unnumbered Interface**.



Item	Description
1	VSX Gateway
2	The external interface serves as the next hop from the Virtual Router
3	External
4	Virtual Router
5	Unnumbered External Interfaces IP "borrowed" from internal interfaces
6	Internal Interfaces with predefined IP addresses
7	Internal

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.
- You can only "borrow" an individual interface IP address once.
- In order to use VPN or Hide NAT, the borrowed address must be routable.

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

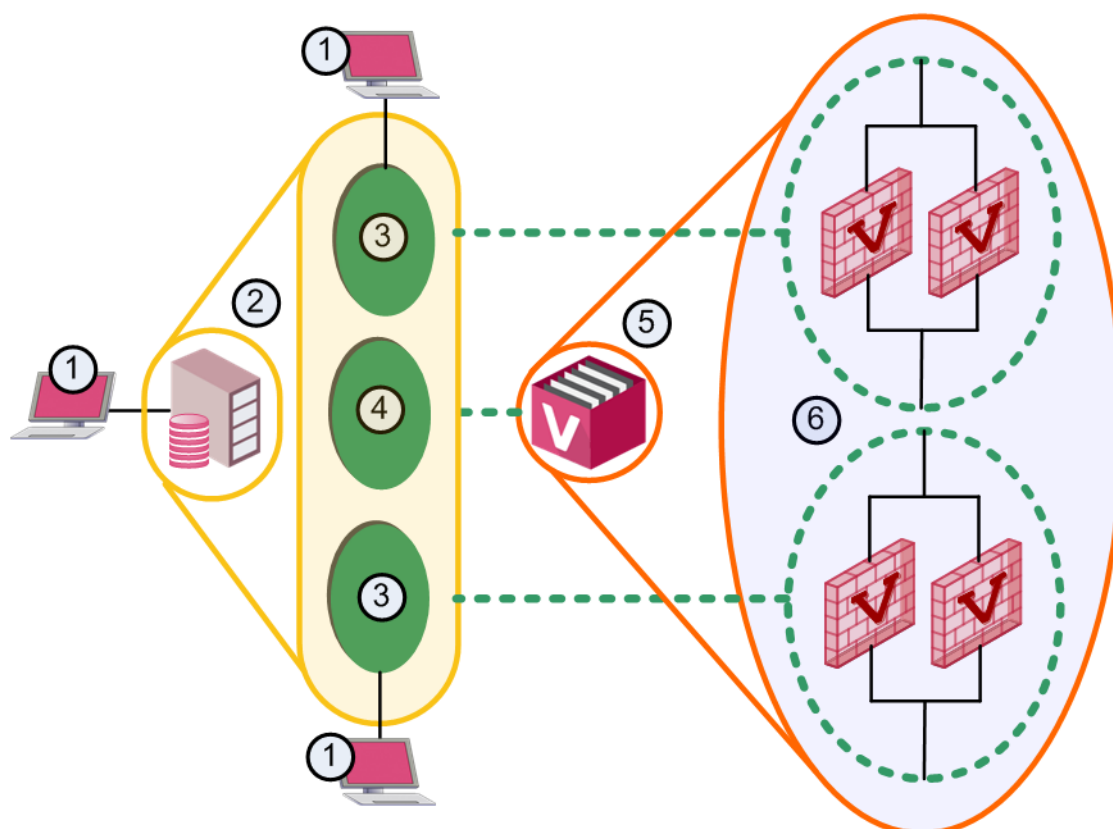
The Security Management Server model is for enterprise deployments with many Virtual Systems but one domain. SmartConsole connects to the VSX Gateway, which contains the Virtual Systems, and directly manages each Virtual System.

Multi-Domain Security Management Model

With Multi-Domain Security Management, you centrally manage multiple networks, typically of different Domains, divisions, or branches. The **Multi-Domain Server** is the central management node that controls the 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.



Item	Description
1	SmartConsole
2	Multi-Domain Server
3	Domain Management Server
4	Main Domain Management Server
5	VSX Gateway
6	Virtual Systems in Domain Management Servers

From a SmartConsole connected to a Multi-Domain Server, provision and configure Domains and Domain Management Servers. Each Domain Management Server uses its own SmartConsole instance to provision and configure its Virtual Systems, virtual devices, and 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.

Feature	Security Management Server	Multi-Domain Security Management (Practical Limit)
Management Domains	1	250
Concurrent Administrators	1	250
Object Databases	1	250
Policies	250	250
Certificate Authorities	1	250
Virtual Systems	25 (recommended)	250

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**.

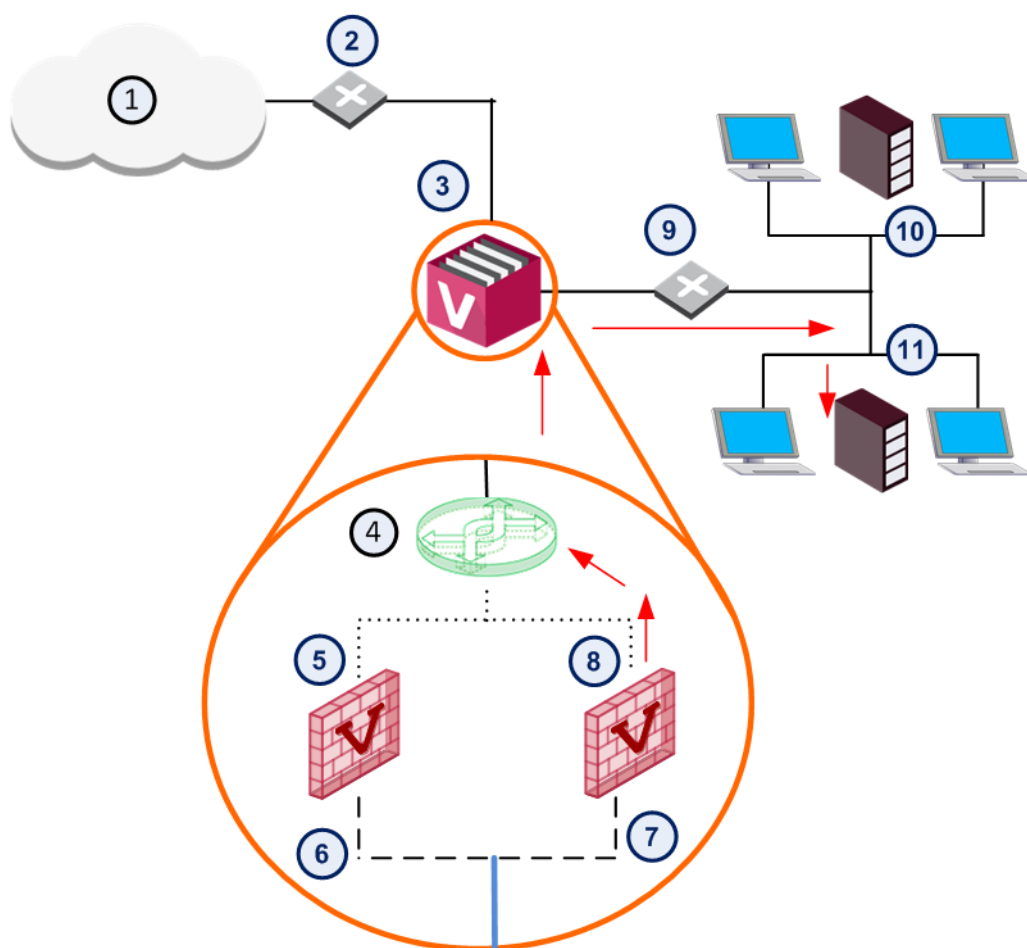
When traffic arrives at a VSX Gateway, a process known as **Context Determination** directs traffic to the appropriate Virtual System, Virtual Router or Virtual Switch. The context determination process depends on the virtual network topology and the connectivity of the virtual devices.

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.



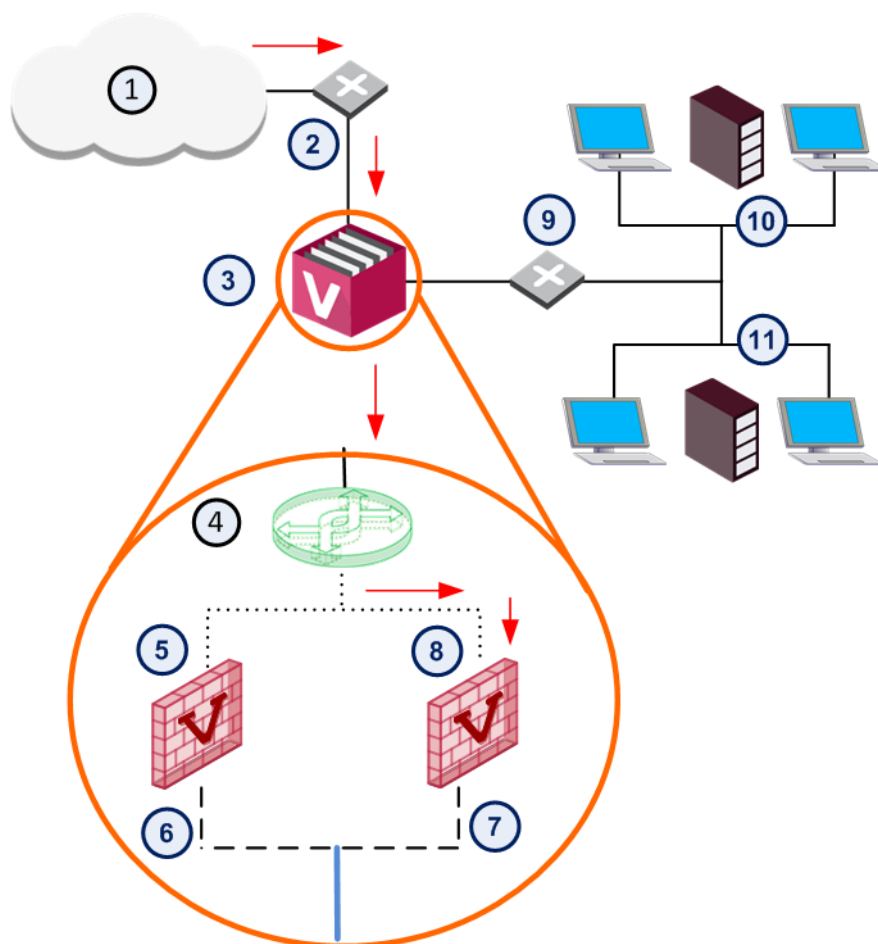
Item	Description
1	Internet
2	Router
3	VSX Gateway </td
4	Virtual Switch
5	Virtual System 1
6	eth1.100
7	eth1.200

Item	Description
8	Virtual System 2
9	VLAN Switch
10	VLAN 100
11	VLAN 200
-----	VLAN Interface
—————	VLAN Trunk
.....	Warp Link

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.



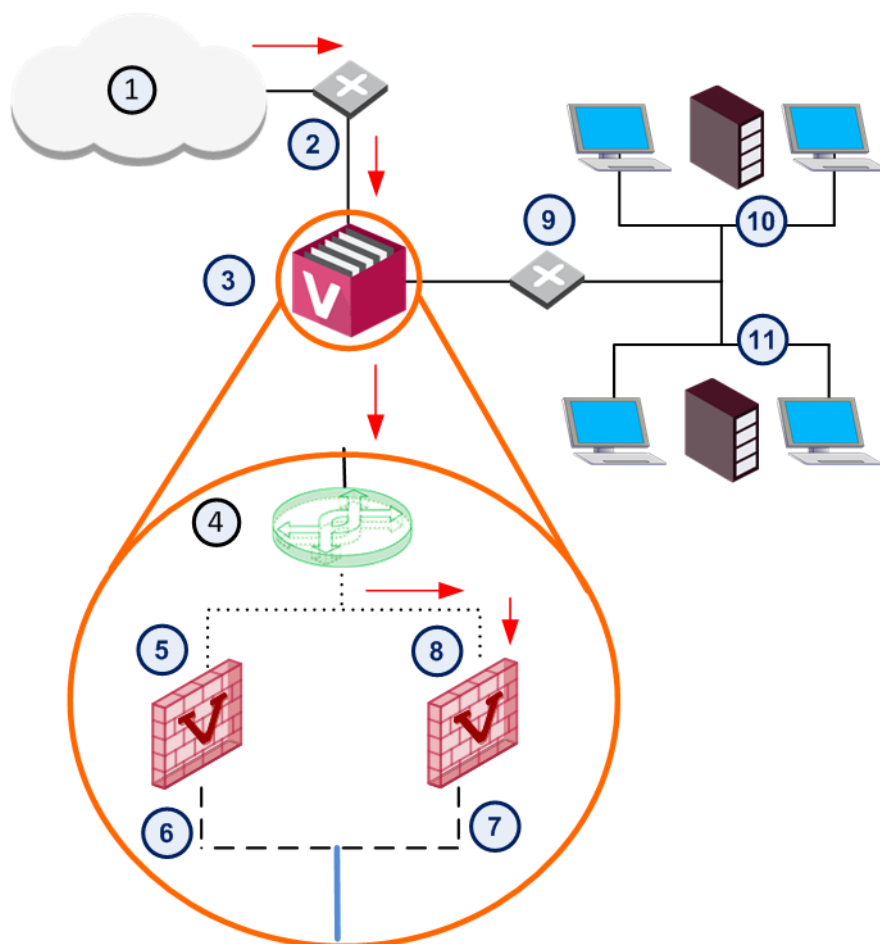
Item	Description
1	Internet
2	Router
3	VSX Gateway
4	Virtual Switch
5	MAC 00:12:C!:Ce:00:01
6	Virtual System 1
7	Virtual System 2

Item	Description
8	MAC 00:12:C!:Ce:00:03
9	VLAN Switch
10	VLAN 100
11	VLAN 200
-----	VLAN Interface
—————	VLAN Trunk
.....	Warp Link

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.



Item	Description
1	Internet
2	Router
3	VSX Gateway
4	Virtual Router
5	172.23.10.11
6	Virtual System 1
7	Virtual System 2

Item	Description
8	172.69.22.30
9	VLAN Switch
10	VLAN 100
11	VLAN 200
-----	VLAN Interface
-----	VLAN Trunk
.....	Warp Link

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

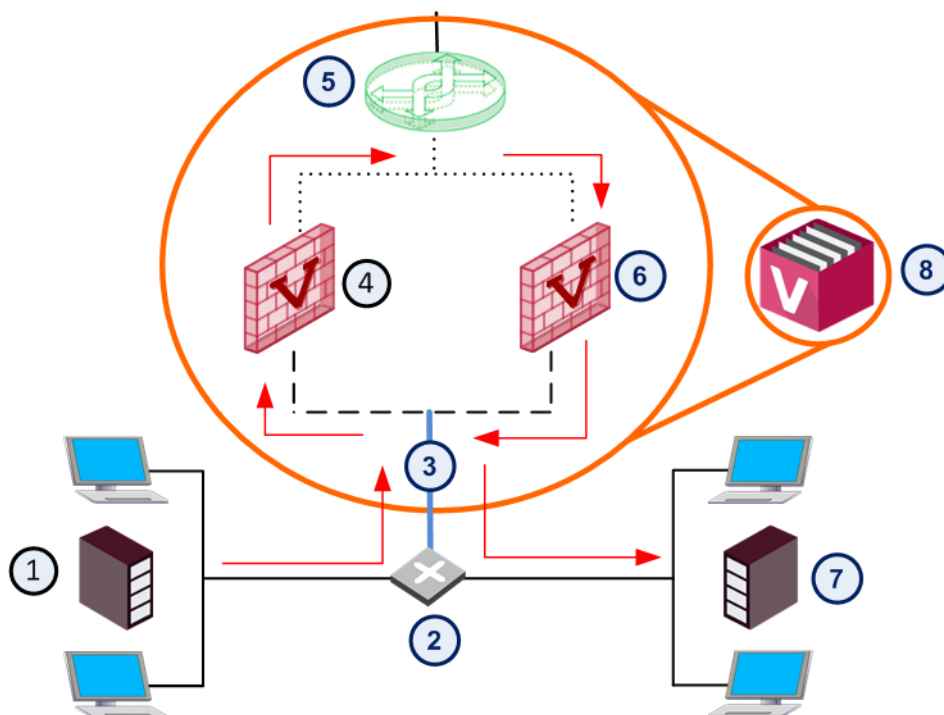
Routing Overview

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.



Item	Description
1	VLAN 100
2	VLAN Switch
3	VLAN Trunk
4	VS 1
5	Virtual Switch
6	VS 2

Item	Description
7	VLAN 200
8	VSX Gateway
----	VLAN Interface
—	VLAN Trunk
.....	Warp Link

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. Based on its routing configuration, VS1 sends the traffic to VS2 by way of the Virtual Switch.
4. VS2 inspects the traffic according to its security policy, inserts a VLAN tag, and sends it to back 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 (wrpj) as the next hop.

Route Propagation using a Virtual Switch

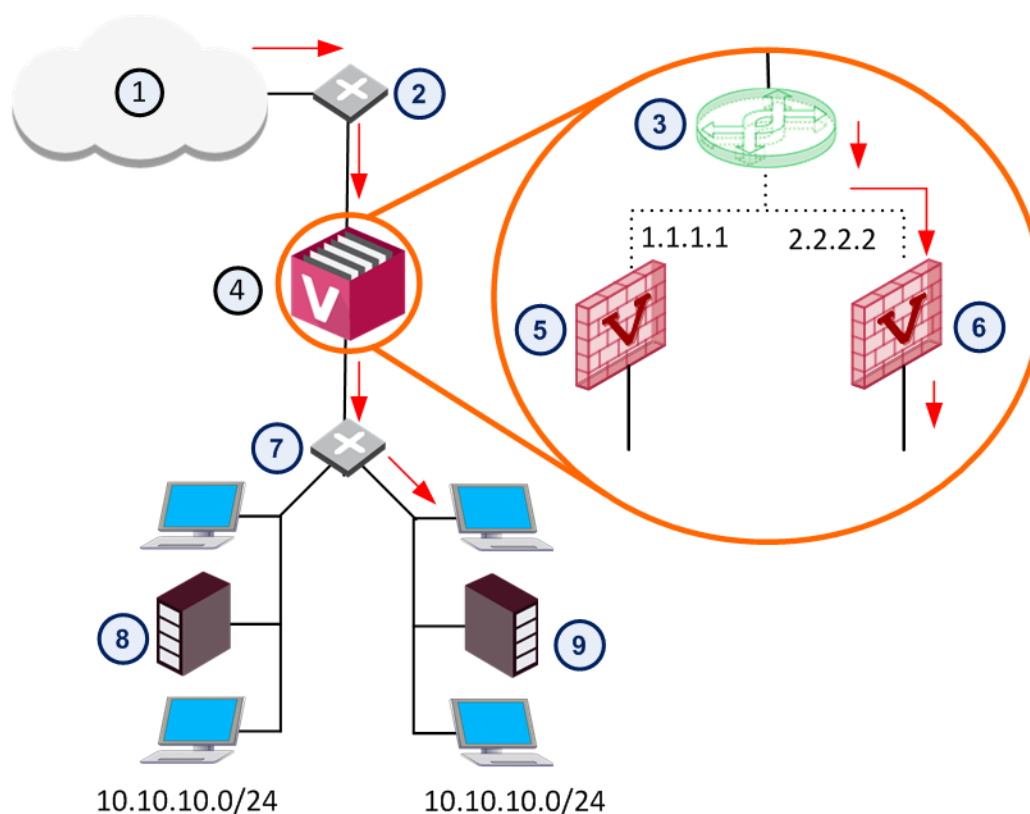
When Virtual Systems are connected to a Virtual Switch, VSX propagates routes by automatically adding entries to the routing table in each Virtual System. Each entry contains a route pointing to the destination subnet using the Virtual System Warp Interface (wrp) IP address.

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.



Item	Description
1	Internet
2	Router
3	Virtual Switch
4	VSX Gateway
5	Virtual System 1

Item	Description
6	Virtual System 2
7	Switch
8	Network 1
9	Network 2
.....	Warp Link

In this case, Network 1 and Network 2 share the same network address pool, which might result in identical overlapping IP addresses. To prevent this, 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.

More for Virtual Switch Route Propagation

You are not required to manually define the topology, because this is done automatically. But there are required manual steps in the VSX objects.

To update the topology map for each Virtual System after you enable route propagation:

1. For each Virtual System object that is connected to the Virtual Switch:
 - a) Edit the object properties. Make sure Anti-Spoofing and VPN features are set correctly.
 - b) Save the object.
2. Install the security policy for the affected Virtual Systems.

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 68) 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 has two or more identical, interconnected VSX Gateways for continuous data synchronization and transparent failover. Virtual System Load Sharing (VSLS) enhances throughput by distributing Virtual Systems, with their traffic load, among multiple, redundant machines.

High Availability

VSX supports High Availability and transparent failover for VSX Gateways and for Virtual Systems. If the active VSX Gateway member fails, all sessions continue to run, securely and without interruption, on a standby cluster member. Users stay connected and do not notice the failover. They are not required to authenticate again on failover.

Use Selective Sync to 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.

Deploying VSX

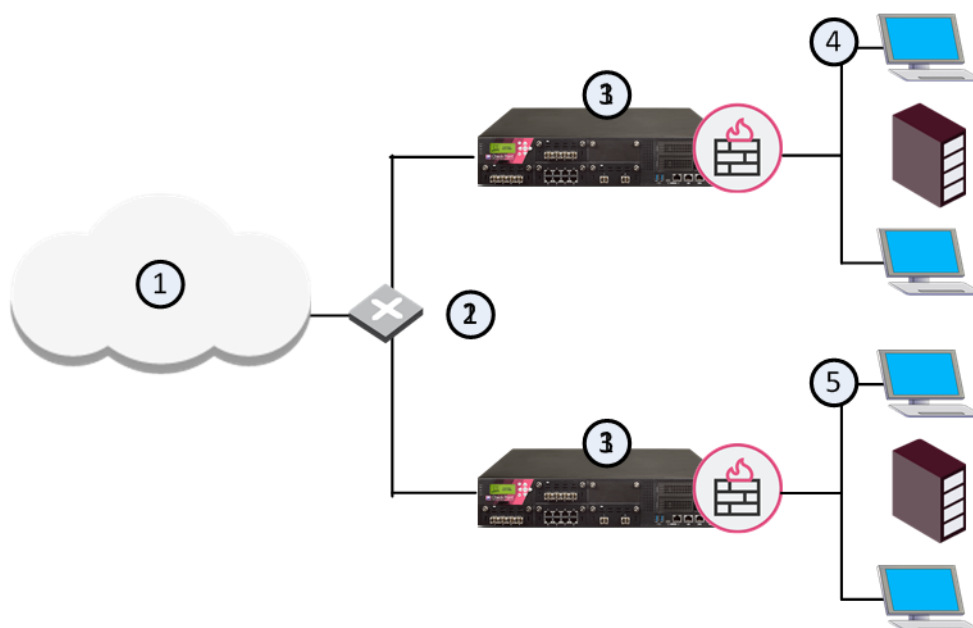
In This Section:

Internal Network Deployment Strategies	40
Organizational Deployment Strategies.....	47
Migrating between Servers with Different Interface Names	56

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, are deployed to protect network segments.



Item	Description
1	Internet
2	Router
3	Security Gateways
4	Network 1
5	Network 2

Each Security Gateway physically connects to its own internal protected network and to a router for access to other internal networks and the Internet.

VSX Virtual System Deployment Strategies

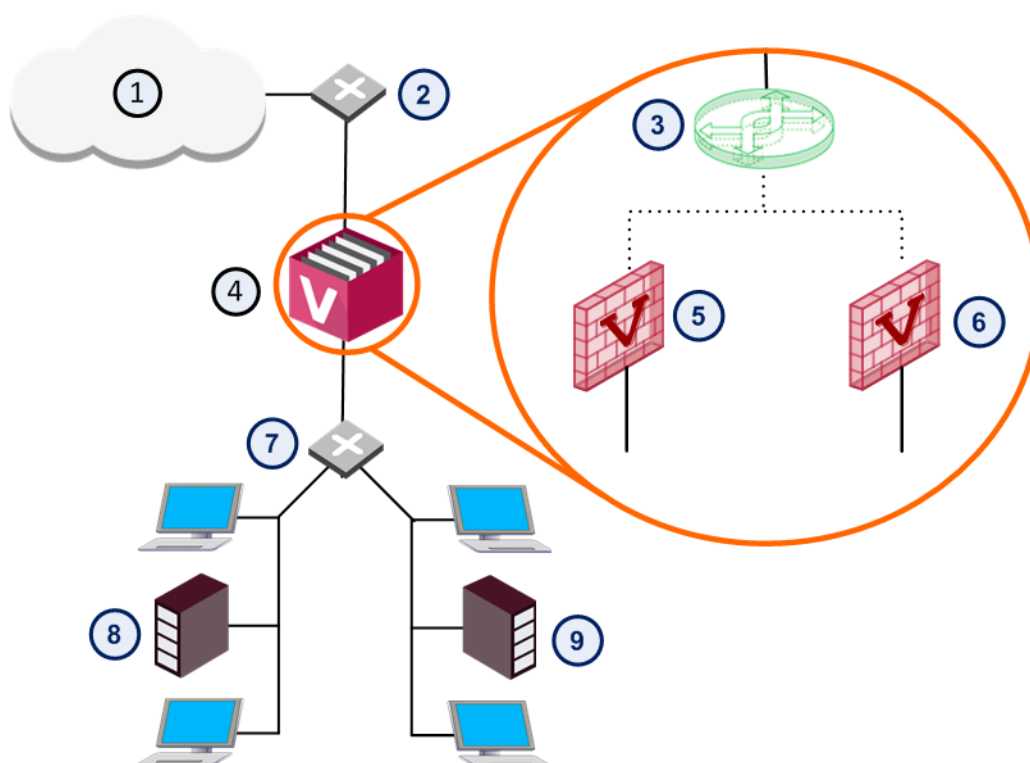
In a VSX environment, Virtual Systems protect internal networks. This section shows sample VSX deployments with Virtual Systems to protect internal networks.

Each example highlights different VSX features. In a real-world deployment, you can combine features to create a powerful cyber security solution for complex enterprise environments.

Physical Internal Interface for Each Virtual System

In a basic VSX configuration, Virtual Systems connect directly to protected internal networks through physical interfaces on the VSX Gateway. A Virtual Switch connects between internal networks, and to the Internet. This deployment 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.



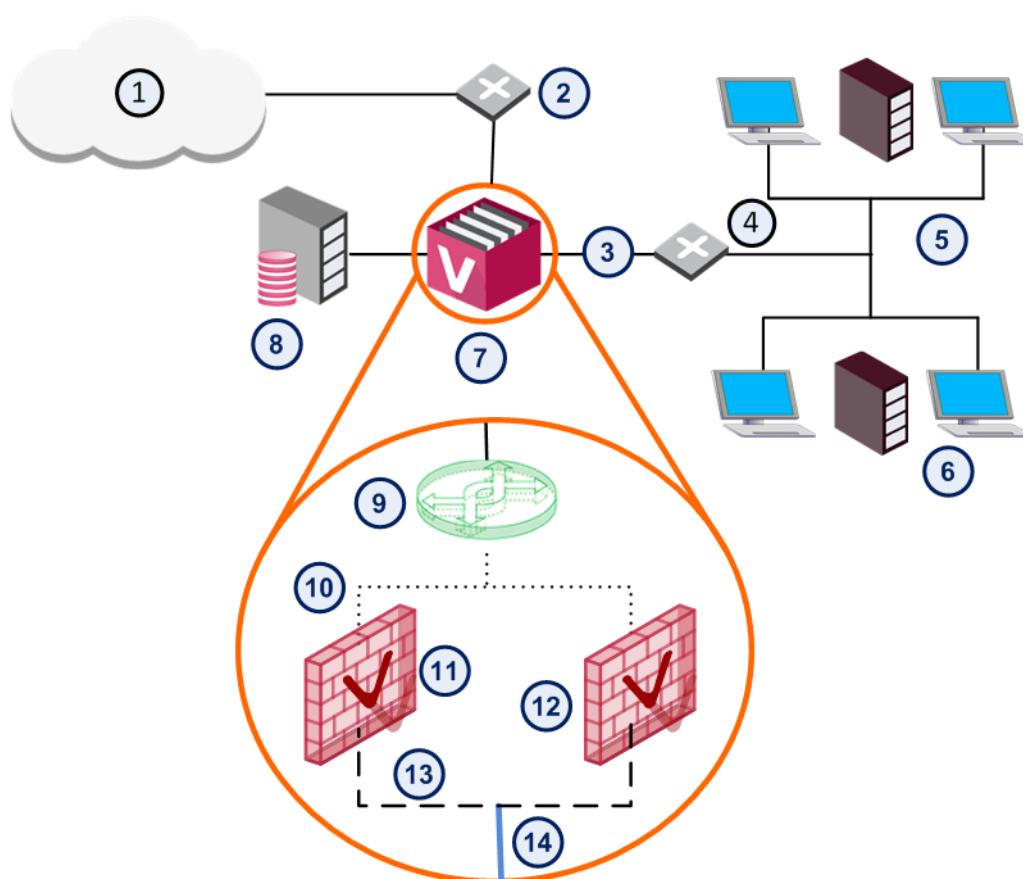
Item	Description
1	Internet
2	Router
3	Virtual Switch
4	VSX Gateway
5	Virtual System 1

Item	Description
6	Virtual System 2
7	Switch
8	Network 1
9	Network 2
.....	Warp Link

Virtual Systems with Internal VLAN Interfaces

In this deployment example, Virtual Systems connect to internal protected networks through VLAN interfaces. The VSX Gateway connects to a VLAN switch with 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 one VSX Gateway or cluster. VLANs provides scalability and granularity, to provision more Virtual Systems and protected networks quickly, without changing the existing IP address structure.



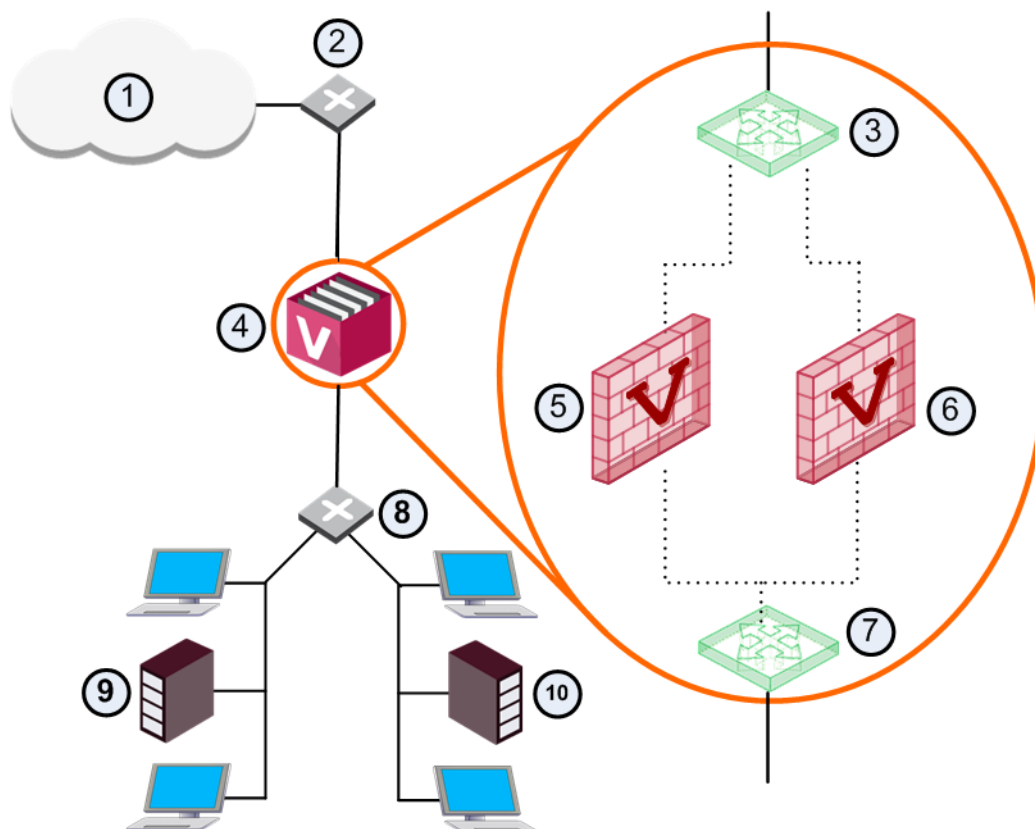
Item	Description
1	Internet
2	Router
3	Physical interface
4	VLAN Switch
5	Network 1
6	Network 2
7	VSX Gateway

Item	Description
8	Security Management Server
9	Virtual Switch
10	Warp interface
11	Virtual System 1
12	Virtual System 2
13	VLAN Interface
14	VLAN Trunk

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.



Item	Description
1	Internet
2	Router
3	Virtual Switch
4	VSX Gateway
5	VS 1
6	VS 2

Item	Description
7	Virtual Router
8	Router
9	Network 1
10	Network 2
.....	Warp link

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 34) provides a detailed discussion of routing options in VSX environments.

Virtual Systems in Bridge Mode

A Virtual System in bridge mode implements native layer-2 bridging instead of IP routing and can co-exist with layer-3 Virtual Systems on the same VSX Gateway. This allows network administrators to easily and transparently deploy a Virtual System in an existing network topology without reconfiguring the existing IP routing scheme.

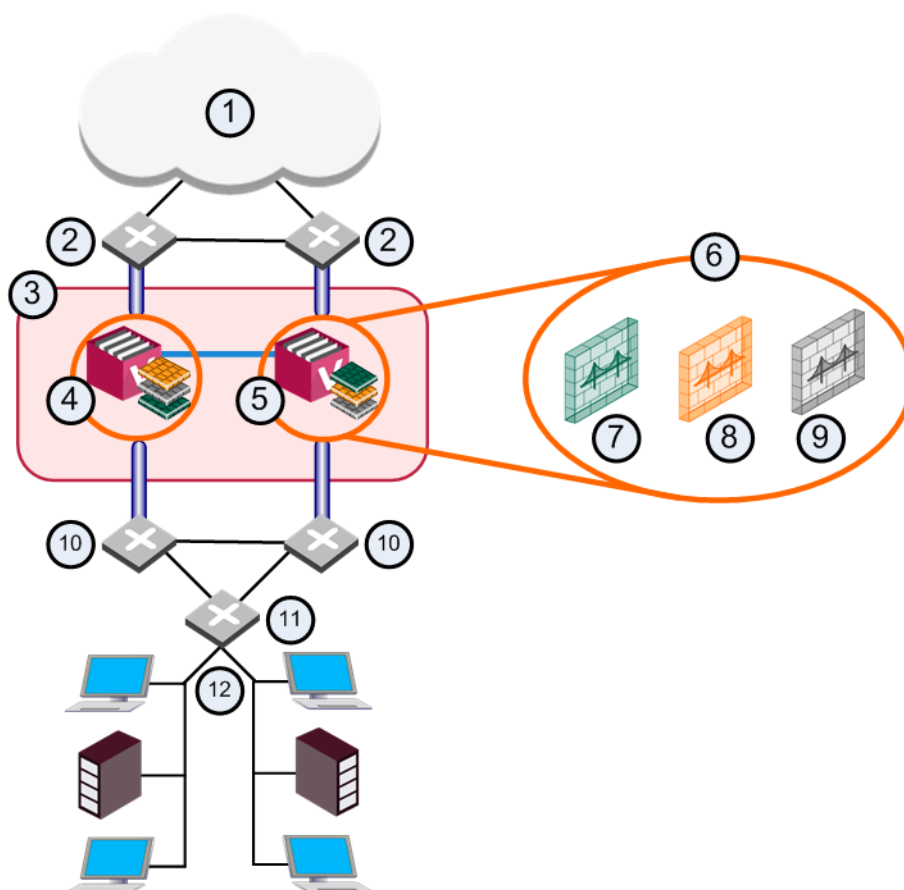
Bridge Mode (on page 113) deployments are particularly suitable for large-scale clustered environments.

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 pair of redundant switches via a VLAN trunk. All Virtual Systems in a given member share the same VLAN trunk.



Item	Description
1	Internet
2	Redundant switches (external)
3	VSX Cluster
4	Member 1
5	Member 2
6	Virtual Systems in Bridge Mode
7	VS 1 Active
8	VS 2 Standby

Item	Description
9	VS 3 Backup
10	Redundant switches (internal)
11	VLAN Switch
12	Internal Networks
	Sync Network
	Physical Interface
	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.

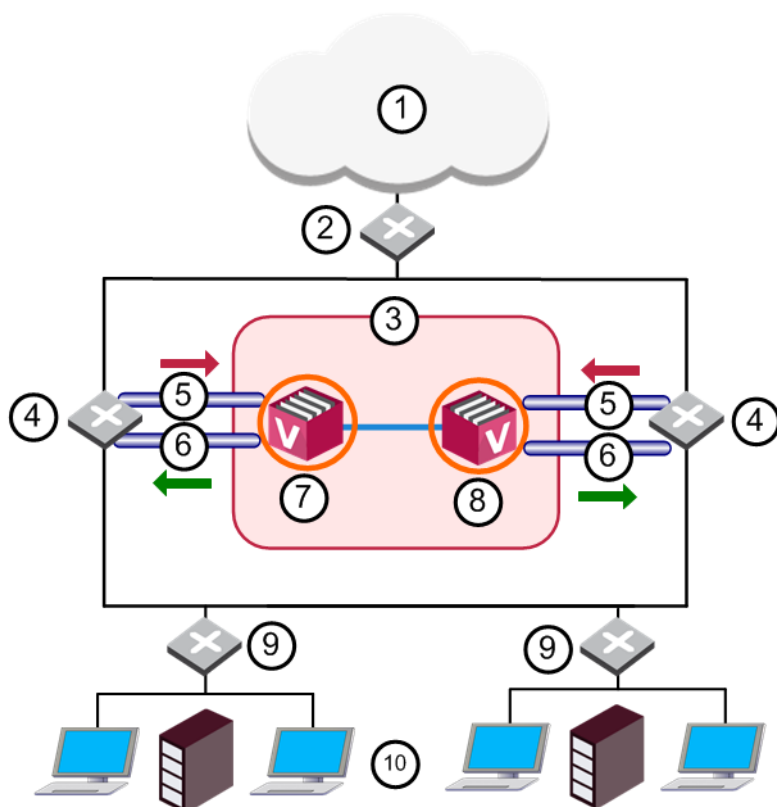
This deployment scenario is appropriate for very large enterprises.

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.



Item	Description
1	Internet
2	Core Network Backbone switch
3	VSX Cluster
4	Router
5	VLAN 10
6	VLAN 20
7	Member 1

Item	Description
8	Member 2
9	LAN switches
10	Internal Networks
— (blue line)	Sync Network
— (black line)	Physical Interface
— (blue oval)	VLAN Trunk

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.

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:

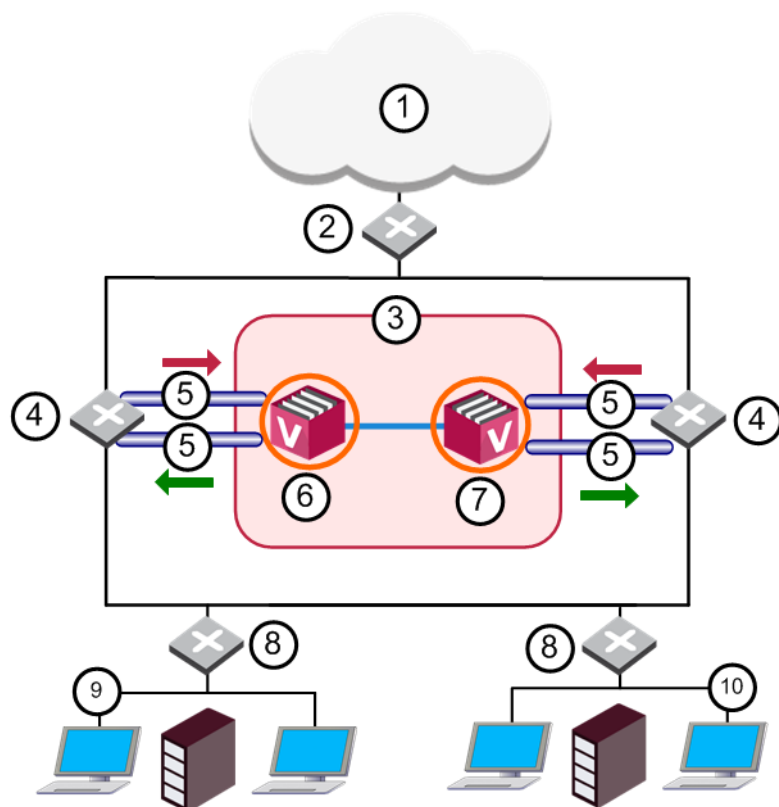
- Large Enterprises
- Managed Service Providers
- Data Centers

Enterprise Deployments

Large enterprise network environments typically have a variety of diverse networks, distributed over multiple locations around the world. These networks often have different security and access requirements for various departments and branches. The ability to centrally manage cyber security, and to maintain 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.



Item	Description
1	Internet
2	Core Network Backbone switch
3	VSX Cluster
4	Router
5	VLAN
6	Member 1
7	Member 2

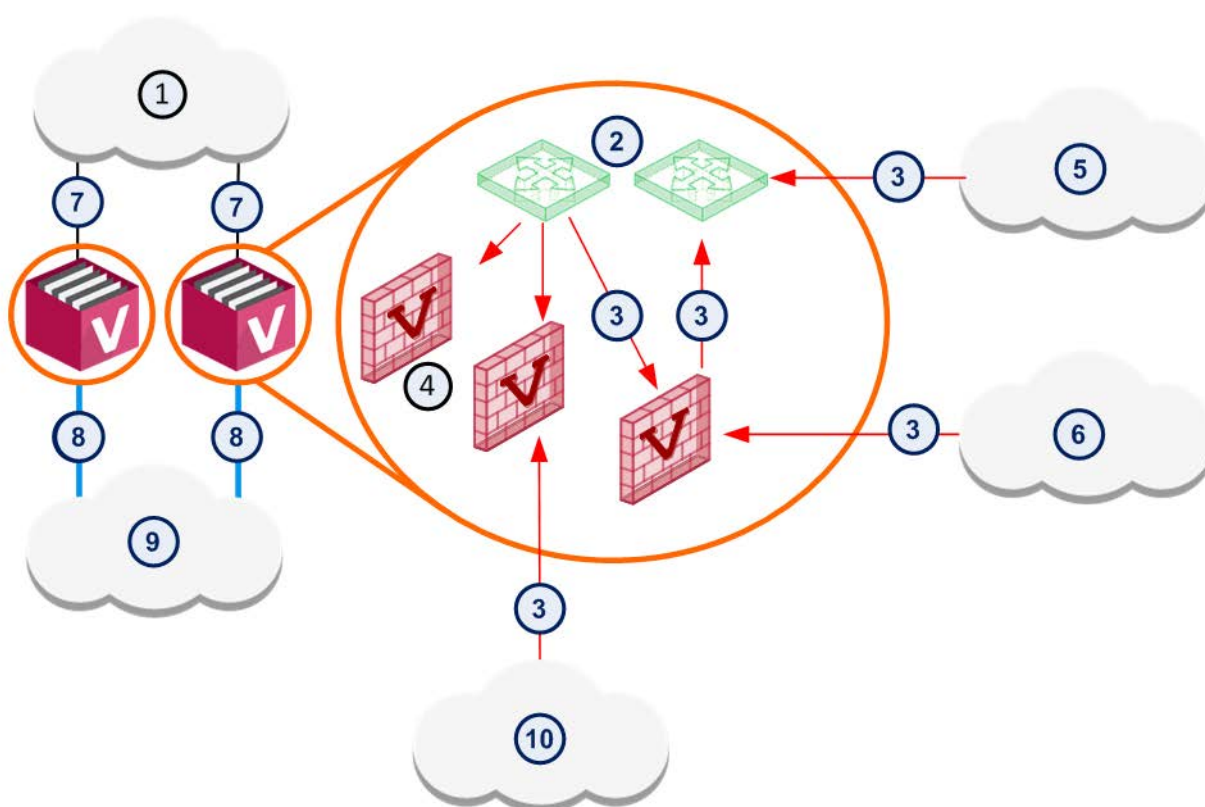
Item	Description
8	LAN Switches
9	Sales
10	Finance
	Sync Network
	Physical Interface
	VLAN Trunk

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.

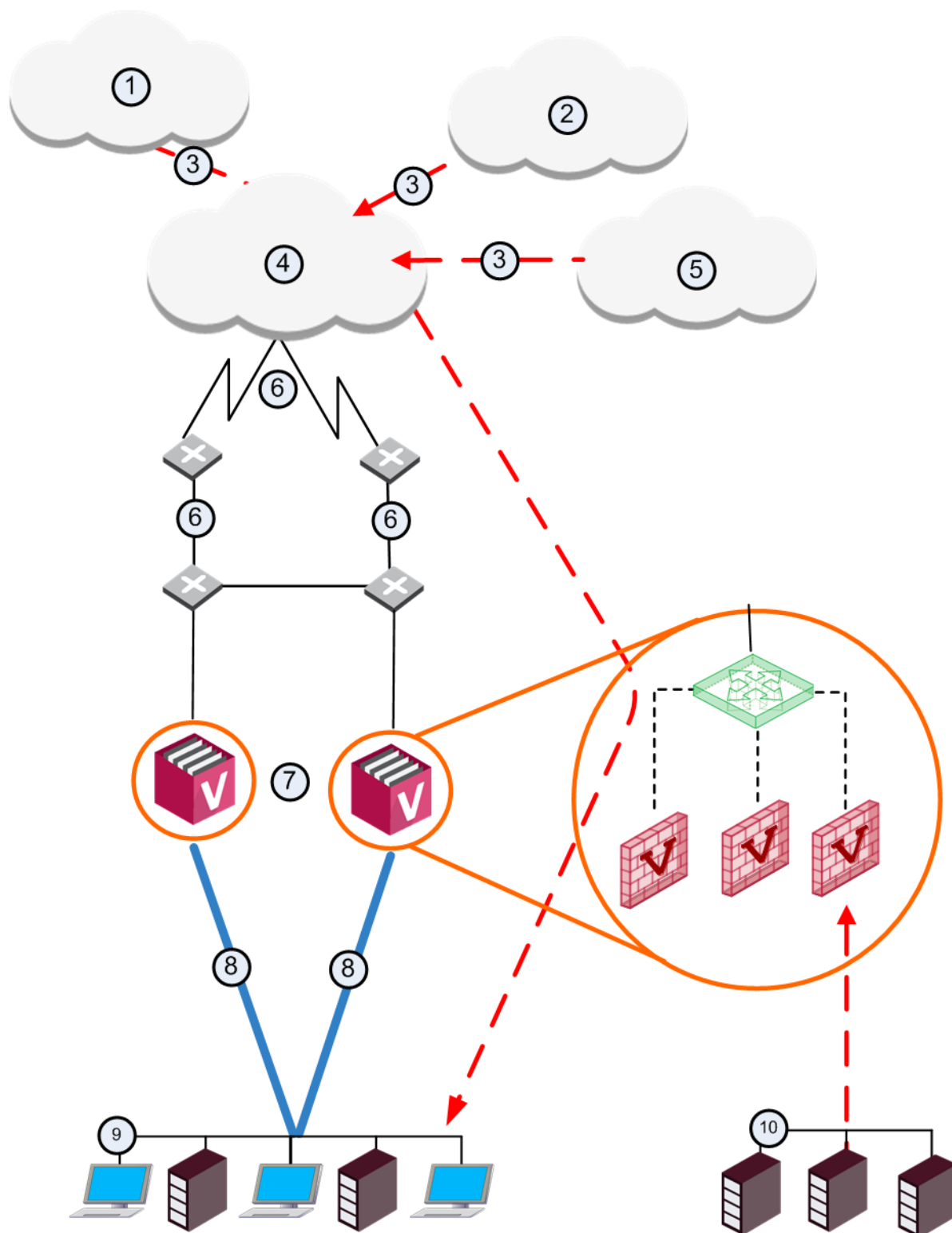


Item	Description
1	Internet
2	Virtual Routers
3	OSPF
4	Virtual Systems
5	Extranet

Item	Description
6	Partner Network
7	BGP
8	OSPF 802.1q
9	BGP in Routed Core
10	DMZ

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.



Item	Description
1	Partner Access
2	Customers
3	IPSec
4	Internet
5	Partner

Item	Description
6	BGP
7	VSX Cluster
8	802.1q VLAN Trunk
9	Internal Network
10	DMZ

Notes to this scenario:

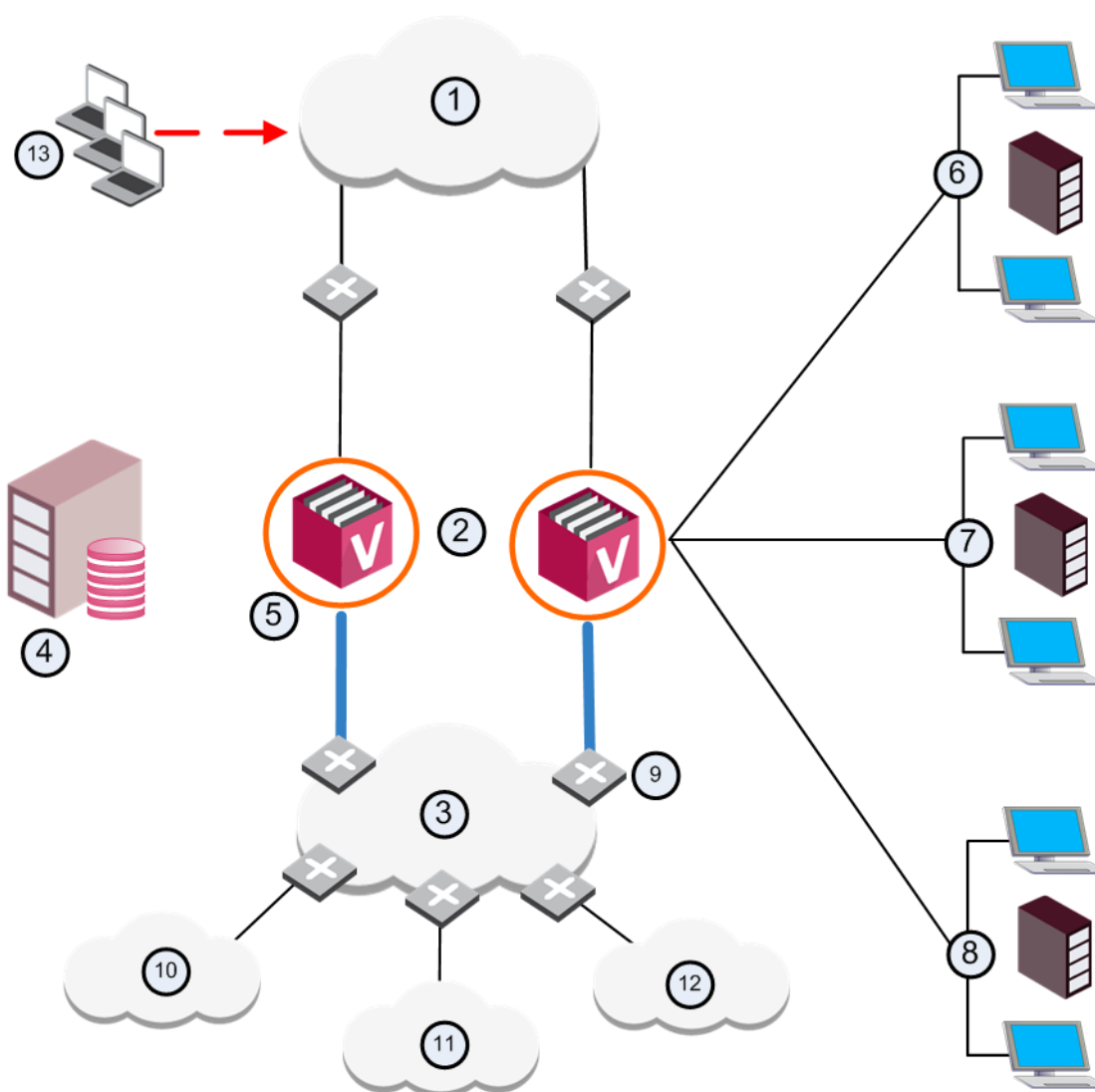
- Partners access network resources remotely via Virtual Systems
- Each Virtual System has its own security policy based on its requirements
- Logs and audit information for each partner is collected separately, and saved to a private database
- Applications and services are segregated by private Virtual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

Item	Description
1	Internet. Routers are between the VSX cluster members and the Internet.
2	VSX cluster. One member handles the Local Exchange and another handles server traffic of different Domains.
3	Core IP VPN Network.
4	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.
5	Multi-Domain Security Management at the NOC and the VSX Gateway make the Local Exchange.
6	Domain A web servers.
7	Domain B DMZ.

Item	Description
8	Domain C mail servers.
9	PE Router.
10, 11, 12	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.
13	Remote access

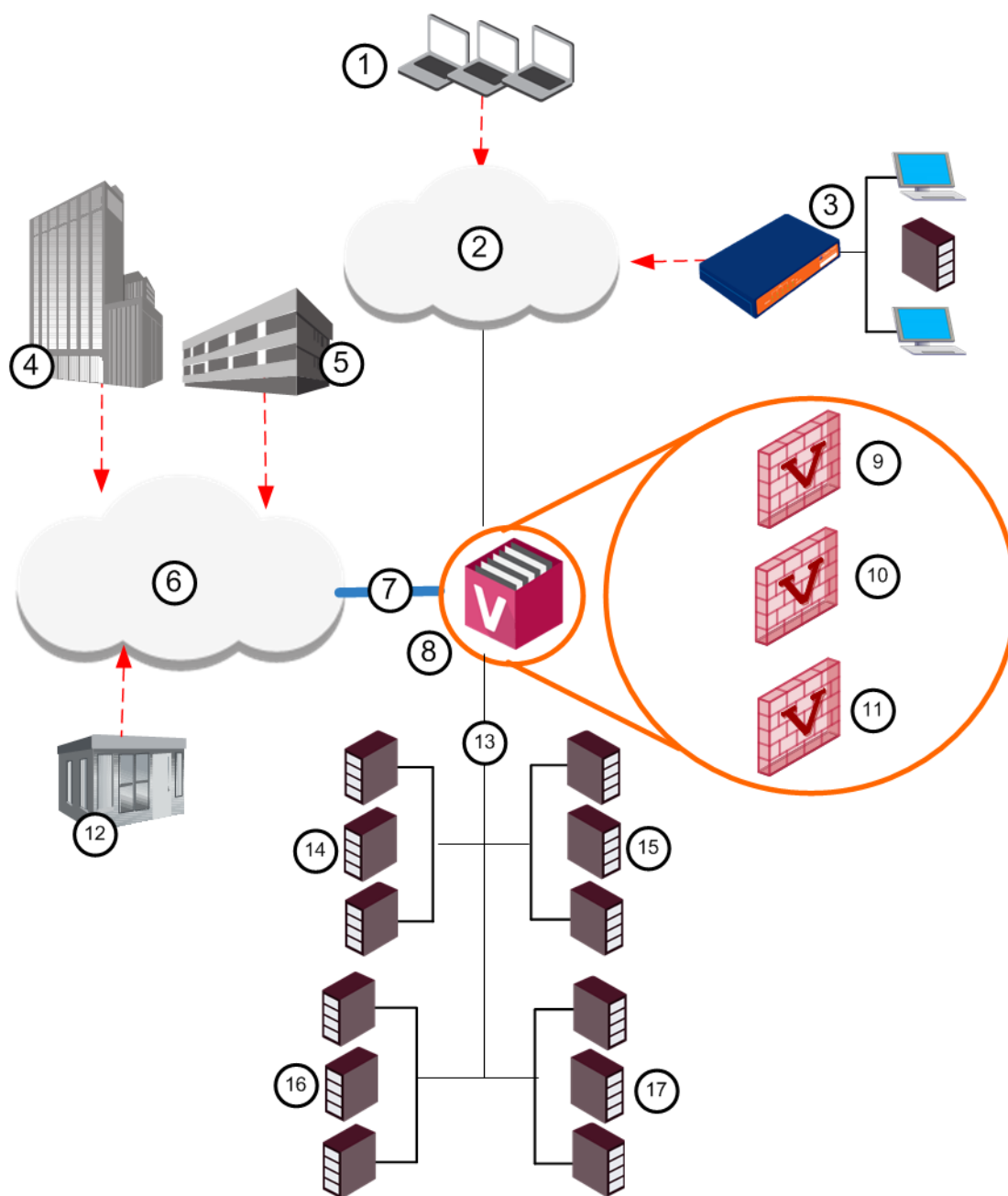
Data Centers

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.

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



Item	Description
1	Remote Users
2	Internet
3	Remote network behind VPN-1 Edge
4	Customer B
5	Customer A
6	MPLS Backbone

Item	Description
10	Virtual System for Customer A
11	Virtual System for Customer C
12	Customer C
13	Data Center
14	Customer A web servers each with separate VLAN ID
15	Customer B mail servers

Item	Description
7	802.1q
8	VSX Gateway
9	Virtual System for Customer A

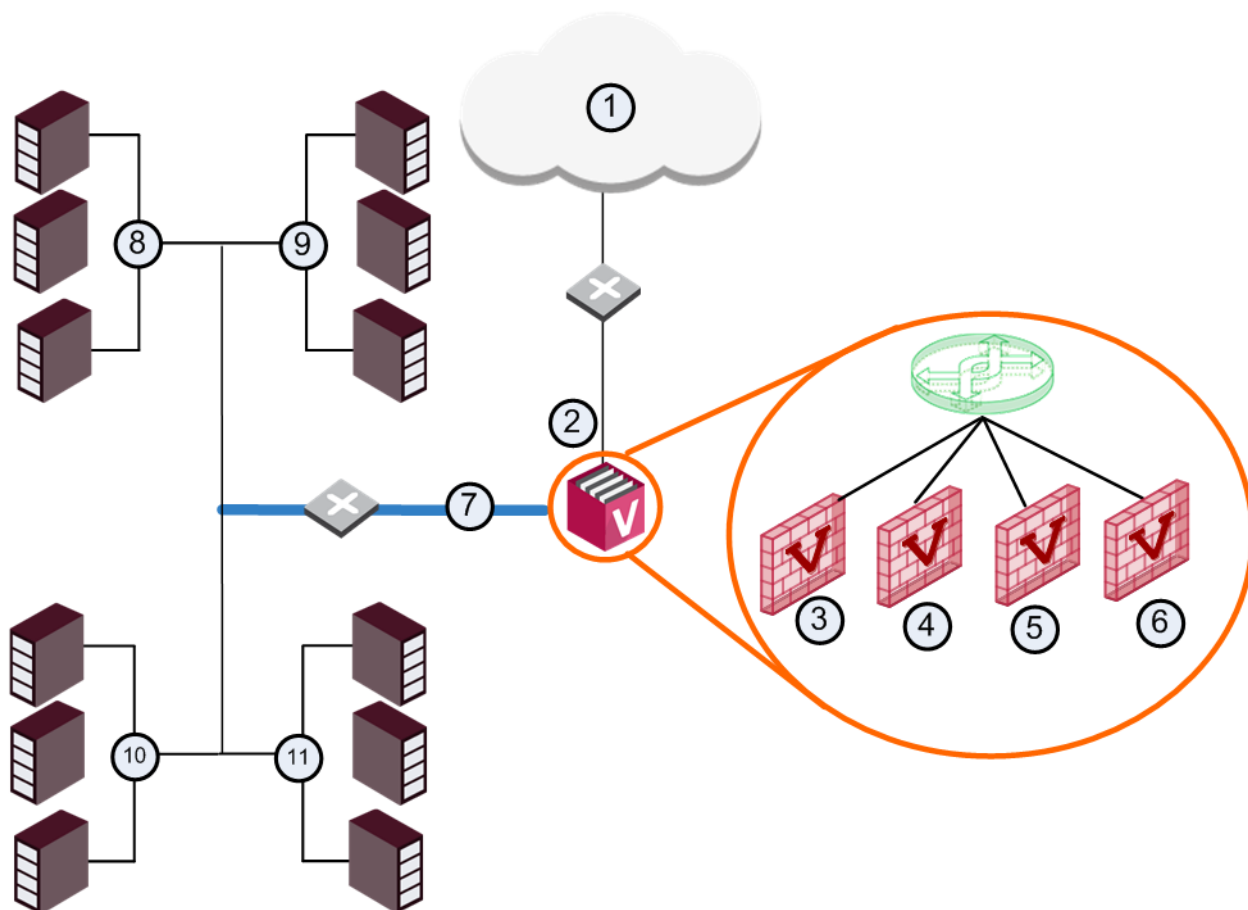
Item	Description
16	Customer A Extranet
17	Customer C databases

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 Check Point appliance that protects a remote network, integrates that network in the MPLS core. A Virtual System can give access to 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.



Item	Description
1	Internet
2	VSX Gateway
3	VLAN 02
4	VLAN 03
5	VLAN 04
6	VLAN 05

Item	Description
7	VLAN Trunk
8	Web Servers VLAN 02
9	Data Bases VLAN 03
10	Application A VLAN 04
11	Application B VLAN 05

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 220) to change the appliance interface names.

The `vsx_util change_interfaces` command ("[change_interfaces](#)" on page 220) 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 224) to push the updated configuration to VSX Gateways or cluster members.

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

1. Close SmartConsole for the Security Management Server and/or Multi-Domain Security Management Domain Management Servers.
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 224) command.
13. Reboot the VSX Gateway or cluster members.

Configuring VSX

In This Section:

Overview	58
Rules & Security Policies	59
Configuring VSX Gateways	59
Working with VSX Gateways	60
Working with Virtual Systems	63
Working with Virtual Switches	69
Creating a New Virtual Switch	71
Working with Virtual Routers	71
CoreXL for Virtual Systems	77
Dynamic Routing for Virtual Devices	79
Working with Interface Definitions	79
Working with Authentication	82
Tracking Activity	89
Working with Network Address Translation	89
Using Application Control and URL Filtering with VSX	90
Using Anti-Bot and Anti-Virus with VSX	90
Licensing VSX	91
Virtual System in Bridge Mode	92

Overview

This chapter explains how to use SmartConsole 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 SmartConsole 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 [100](#)) explains these procedures.

To configure virtual devices, make sure:

- The management servers (Security Management Server or Multi-Domain Server) are configured and running.
- SmartConsole is installed on the appropriate computer.

This chapter assumes that you are familiar with SmartConsole 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

You use the same procedures to define and install security policies on a VSX gateway or Virtual System as for a physical Security Gateway. This statement also applies to the use of IPv6 in security policies. These procedures are not included 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 SmartConsole when you install a policy.

Configuring VSX Gateways

Configuring 64-Bit Virtual System Support

You can configure a Security System to run fwk as a 64 bit process. This lets VSX Virtual Systems use more than 4 GB of RAM, which significantly increases the concurrent connection capacity for each Virtual System.

Use the `vs_bits` command to configure fwk to run in the 64 or 32 bit mode. The system automatically reboots when you run the command.

Important: Run the `vs_bits` command only from a VS0 context.

Syntax:

```
vs_bits [-stat | 32 | 64 ]
```

Parameter	Description
stat	Shows the current fwk mode
32	Run fwk in the 32 bit mode
64	Run fwk in the 64 bit mode

Examples:

This example changes the fwk mode to 64 bits:

```
vs_bits 64
```

This example shows the fwk modes:

```
# vs_bits -stat
All VSs are at 64 bits
```

Known limitations:

- This feature only works on a 64 bit operating system.
- The VSX gateway will automatically run `cpstop;cpstart`.
- To change to 64-bit mode in a cluster, use the Connectivity Upgrade procedure.

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 SmartConsole. 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. In **Gateways & Servers** view or **Object Explorer**, double-click the VSX Gateway.
You can also search for the VSX Gateway in the Object Explorer.
2. In the **VSX Gateway Properties** window, click **Communication**.
3. In the **Trusted Communication** window, enter and confirm the SIC Activation Key.
4. 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 79).
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, change, and delete 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 Gateways & Servers view or Object Explorer tree, right-click the VSX Gateway object on the Object Tree and select **Delete**.
2. In the window that opens, click **Yes**.

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 and its virtual device configuration.

To restore a VSX Gateway configuration:

1. Install the gateway again.
2. Configure the IP address, net mask, and default gateway.
3. Make sure all management interfaces have the same IP addresses as before.
4. 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

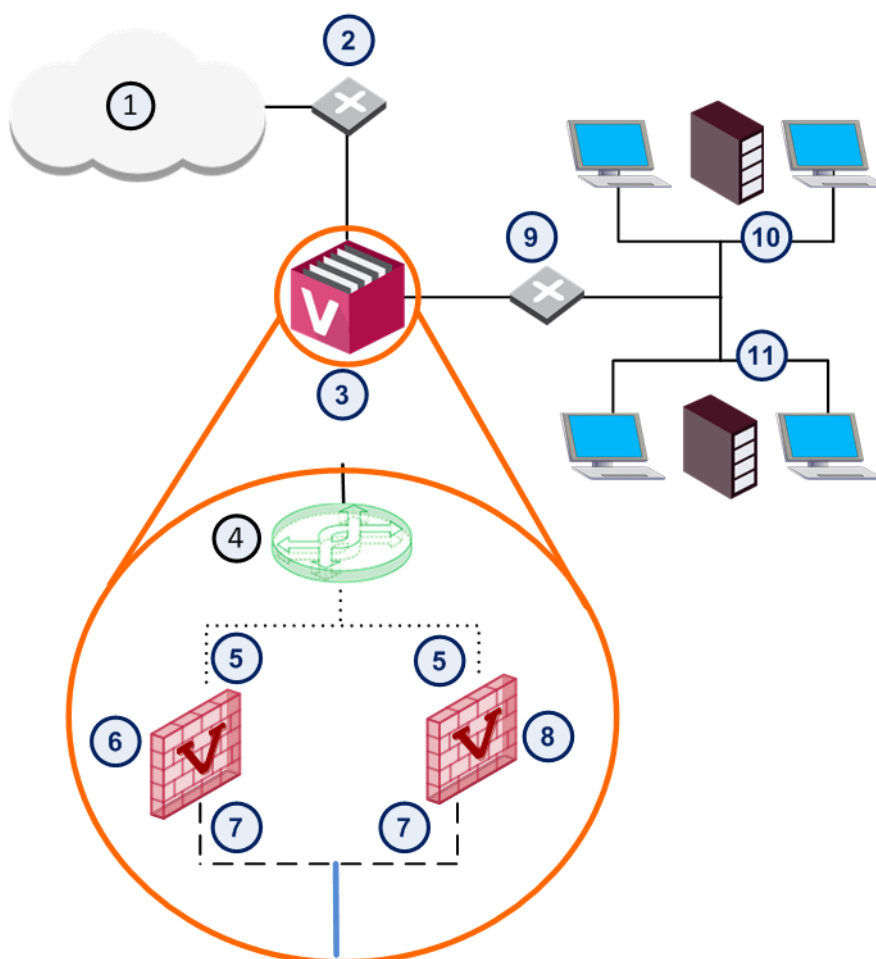
VSX supports up to 128 interfaces for each virtual device and a total of up to 4096 interfaces per VSX Gateway or cluster. The supported interfaces include VLANs and Warp Links.

Note - By default, a Virtual System supports up to 64 interfaces. For more about how to increase the number of supported interfaces, see sk99121

<http://supportcontent.checkpoint.com/solutions?id=sk99121>.

You can add as many interfaces to a Virtual System as required, according to system resources.

Here is an example of a typical VSX Gateway deployment with two Virtual Systems, each with two interfaces.



Item	Description
1	Internet
2	Router
3	VSX Gateway
4	Virtual Switch
5	External Interface
6	Virtual System 1
7	Internal Interface

Item	Description
8	Virtual System 2
9	VLAN Switch
10	Network 1
11	Network 2
-----	VLAN Interface
-----	VLAN Trunk
.....	Warp Link

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 SmartConsole.
If you are using Multi-Domain Security Management, open SmartConsole from the Domain Management Server in which you are creating the Virtual System.
2. 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

In the **Virtual System Network Configuration** page, define internal and external interfaces and the IP address topology behind the internal interface. The process to define Virtual System network properties is different in different environments:

- Use the **VSX Gateway Creation** template to define the VSX Gateway that contains the Virtual System.
- If you choose to override the default VSX Gateway Creation template, you can use the **Custom Configuration** template.
- You can create the Virtual System in 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 67).

Separate Interfaces in Bridge Mode

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

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:
 - 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. To create a new IP address definition:
 - a) Select **Specific**, and click **New**.
 - b) Select **Group** to define an object group, or **Network** to define network properties.
4. Enable **Layer-3 bridge interface monitoring** to enable layer 3 network fault detection for this Virtual System.
Enter an IP address and subnet mask, which continuously monitors the specified network for faults or connectivity issues. The IP address/subnet define the network on which the Virtual System resides.
5. Complete the definition process ("[Completing the Definition](#)" on page 67).

Custom Configuration or Override - Non-Bridge Mode

If you used the Custom Configuration template when creating the VSX Gateway, or if you selected **Override Creation Template**, manually define the network interfaces and connections. The **Virtual System Network Configuration** page for Custom Configuration opens.

To configure the external and internal interfaces:

1. In the interface table, define the applicable interfaces.
You can add new interfaces and delete and change existing interfaces.
To add an interface, click **Add**. The **Interface Properties** window opens. Select an interface from the list and define its properties.
2. Select the **Main IP Address** from the list.
This IP address is usually assigned to the external interface and specifies the Virtual System address used with NAT or VPN connections.
To make an external IP address routable, select the external interface IP address as the main IP address.
3. Define network routing for your deployment.
Some routes are automatically defined by the interface definitions. For example, you define a default gateway route leading to an external Virtual Router or to the Virtual System external interface.
To manually add a default route to the **Routes** table, click **Add Default Routes**. Enter the default route IP address, or select the default Virtual Router. The **Route Configuration** window opens.
4. Complete the definition ("[Completing the Definition](#)" on page 67).

Custom Configuration or Override in Bridge Mode

If you used the **Custom Configuration** template to create the VSX Gateway, or if you selected the **Override Creation Template** option for a Virtual System in Bridge Mode, manually define the network interfaces.

Interfaces: To configure the external and internal interfaces, define interfaces and links to devices in the **Interfaces** table. You can add, change, and remove interfaces. To add an interface, click **Add**. The **Interface Properties** window opens. Select an interface from the list and define its properties.

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 187) for further assistance.

After you create a Virtual System using the Virtual System Wizard, you can modify the topology and all other parameters (except the name of the Virtual System) using the **Virtual System Properties** window.

Modifying a Virtual System

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

To change a Virtual System, 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 79) 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 61).
- **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 belong 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

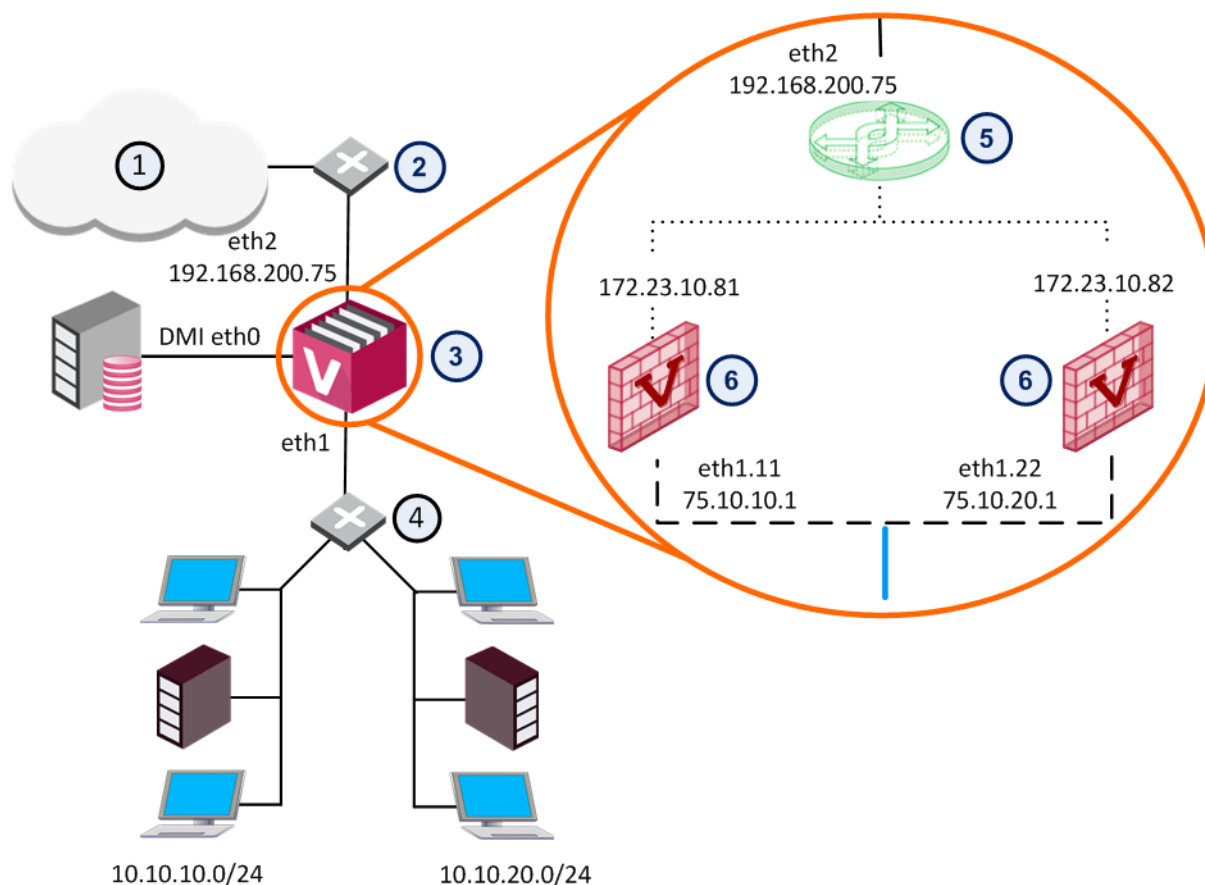
To delete a Virtual System:

1. From the Gateways & Servers view or Object Explorer tree, right-click the Virtual System object and select **Delete**.
2. In the window that opens, click **Yes**.

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.



Item	Description
1	Internet
2	Router
3	VSX Gateway
4	VLAN Switch
5	Virtual Switch

Item	Description
6	Virtual Systems
-----	VLAN Interface
-----	VLAN Trunk
.....	Warp Link

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

Modifying a Virtual Switch

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

To modify a Virtual Switch, 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 SmartConsole.

Virtual Switch - Topology

The **Topology** page defines Virtual Switch interfaces. You can only modify the single defined interface. You cannot change the settings for Warp interfaces in this window.

To add an interface:

1. Click **New**.
The **Interface Properties** window opens.
2. Select an interface from the list and define the IP address, net mask and other properties.
3. **Optional:** Click **Actions > Copy to Clipboard** to copy the **Interfaces** table in CSV format.

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 Gateways & Servers view or Object Explorer tree, right-click the VSX Gateway object on the Object Tree and select **Delete**.
2. In the window that opens, click **Yes**.

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 SmartConsole.

If you are using Multi-Domain Security Management, open SmartConsole from the Domain Management Server in which you are creating the Virtual Switch.

2. From the **Objects Bar (F11)**, click **New > More > Network Object > Gateways and Servers > 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**.

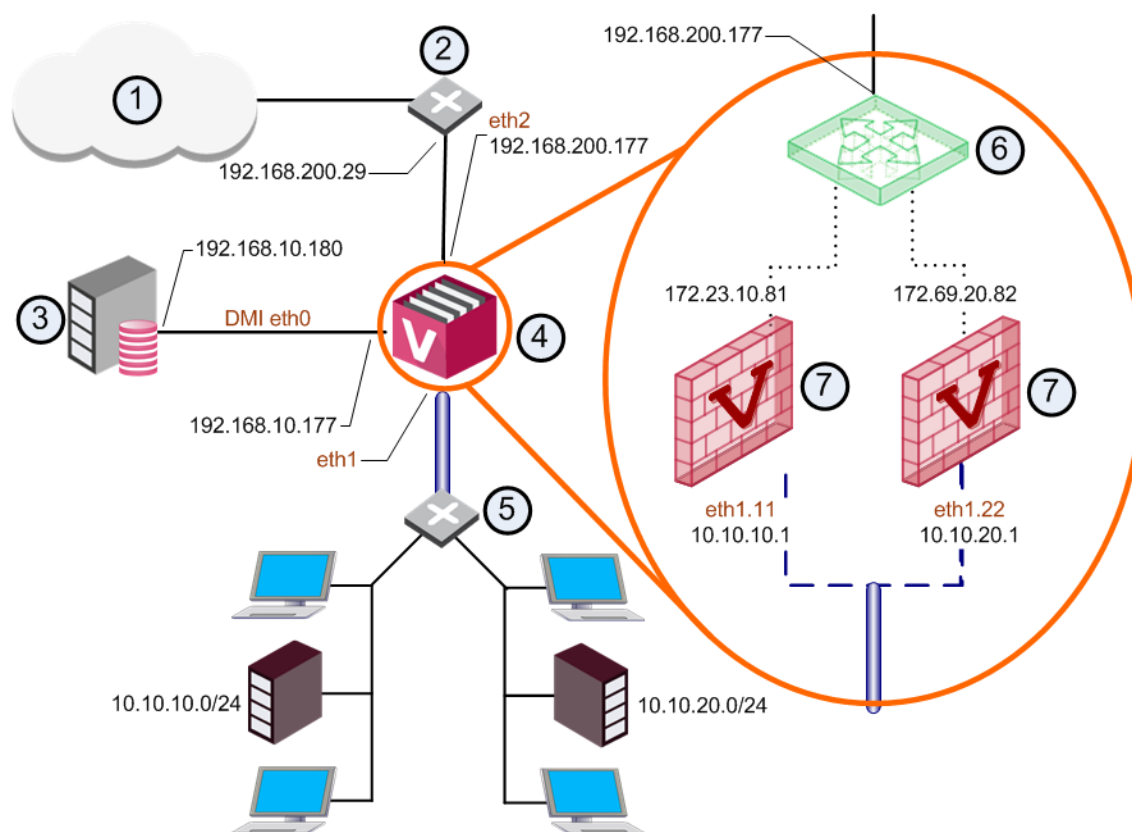
The **Add Interface** window opens.

7. In the **Add Interface** window, 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.

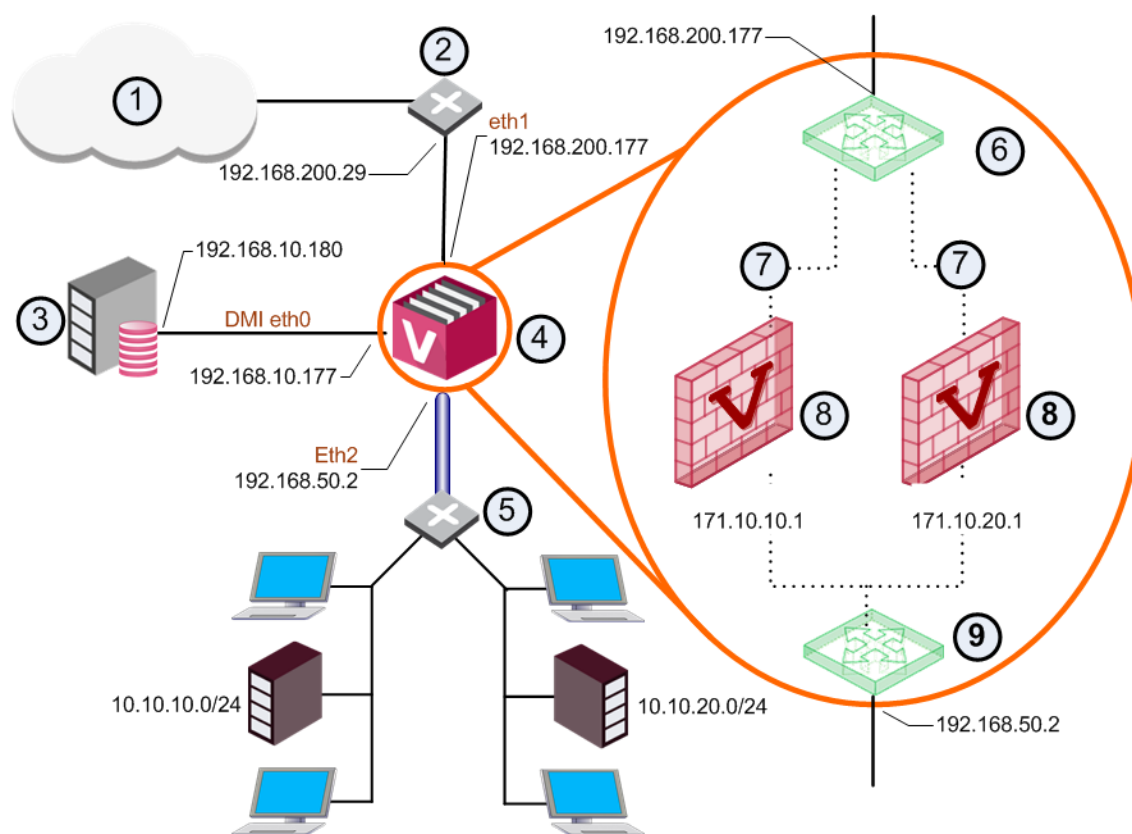
You can define Virtual Routers for both external and internal communications. A Virtual Router that connects to external networks, including a DMZ and the Internet, are referred to as an external Virtual Router. A Virtual Router that connects to internal, protected networks is known as an internal Virtual Router.



Item	Description
1	Internet
2	Router
3	Security Management Server
4	VSX Gateway
5	VLAN switch

Item	Description
6	External Virtual Router
7	Virtual Systems
----	VLAN Interface
—	VLAN Trunk
.....	Warp Link

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.



Item	Description
1	Internet
2	Router
3	Security Management Server
4	VSX Gateway
5	Switch
6	External Virtual Router

Item	Description
7	Unnumbered
8	Virtual Systems
9	Internal virtual router
----	VLAN Interface
—	VLAN Trunk
.....	Warp link

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 SmartConsole.
If you are using Multi-Domain Security Management, open SmartConsole from the Domain Management Server in which you are creating the Virtual System.
2. From the Gateways & Servers view or Object Explorer, 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 SmartConsole.

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 79).
- **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 75) rules.

Deleting a Virtual Router

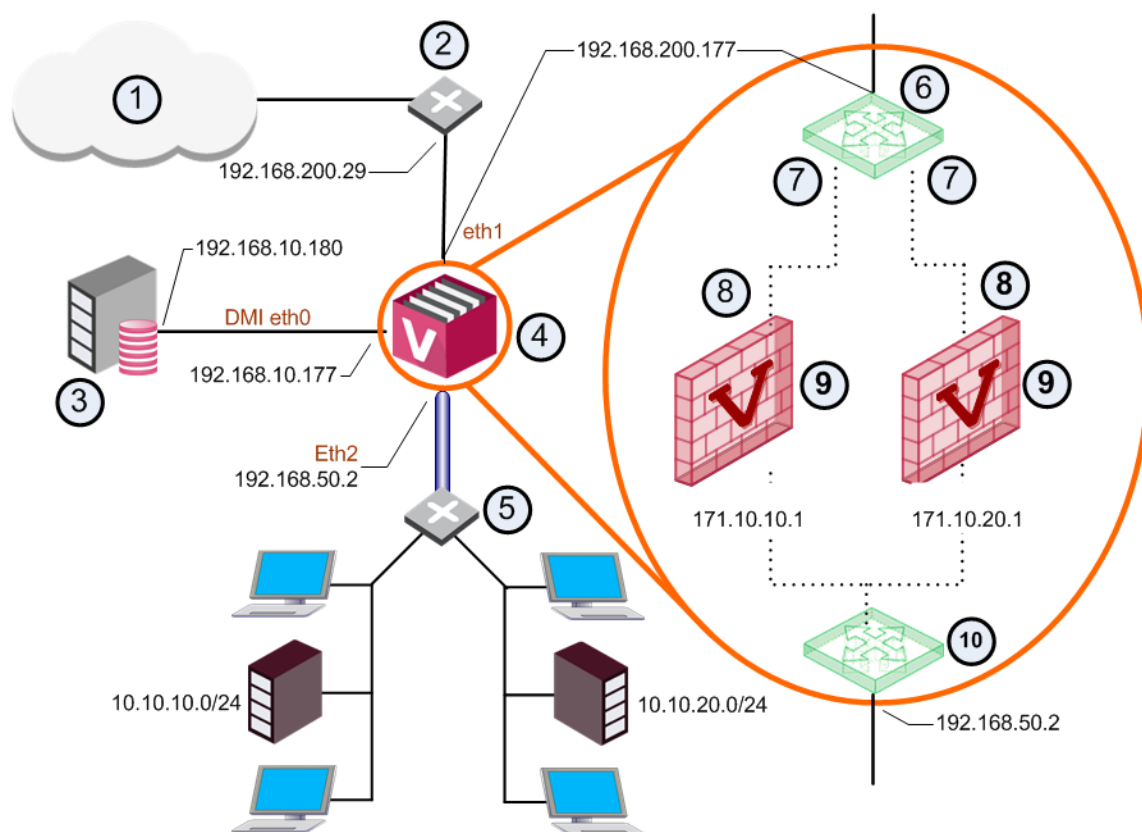
You cannot delete a Virtual Router if it is still connected to a Virtual System. Remove all Virtual Router connections before deleting.

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.



Item	Description
1	Internet
2	Router
3	Security Management Server
4	VSX Gateway
5	Switch
6	External Virtual Router
7	wrpj

Item	Description
8	wrp unnumbered
9	Virtual Systems
10	Internal virtual router
----	VLAN Interface
—	VLAN Trunk
.....	Warp link

Defining Source-Based Routing Rules

Define Source-based Routing rules in the **Topology** page of the Virtual Router definition window.

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 SmartConsole.
2. From the Gateways & Servers view or Object Explorer, 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 205).

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 SmartConsole 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 *R80.10 Performance Tuning Administration Guide* <http://downloads.checkpoint.com/dc/download.htm?ID=54765>.

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 SmartConsole 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 can have a negative 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 SmartConsole.
2. From the Gateways & Servers view or Object Explorer, 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.



Important - You cannot use the CLI to configure static routes for VSX. You can only configure them in SmartConsole.

For more about configuring dynamic routing, see the *R80.10 Gaia Advanced Routing Administration Guide* <http://downloads.checkpoint.com/dc/download.htm?ID=54803>.

To configure dynamic routing for a virtual device:

1. Set the context to the virtual device. From the Gaia 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 36) 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** for IPv4 and **/128** for IPv6.
- **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 36) provides additional details.
- **MTU:** Maximum transmission unit size in bytes (default = 1,500). The minimum and maximum MTU values are:
 - IPv6 MTU: 1280 – 16000
 - Ipv4 MTU: 68 – 16000

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

Attackers 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.

	From	To
IPv4 (defined in RFC 1112)	224.0.0.0	239.255.255.255
IPv6	ff00::	ffff:ffff:ffff:ffff:ffff:ffff:ffff:fff f

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 a restriction type:
 - **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 79).

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 *R80.10 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 of 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>.

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:

1. Configure shared authentication on the Virtual Systems.
 - a) Open SmartConsole.
 - b) From the Gateways & Servers view or Object Explorer, 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
`# vsend <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**.
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 SmartConsole, 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 `$VAR_ACE`.
3. Copy the `securid` file to `$VAR_ACE`.
For VS0, copy the file to `/var/ace`.
4. For cluster configurations, for each cluster member:
 - Locate a Virtual System that is active on that member and do the all the previous steps.
 - If there are no active Virtual Systems on that member, fail-over to the cluster member and then do the 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:

1. Configure private authentication on the VSX Gateway and the Virtual Systems.
 - a) Open SmartConsole.
 - 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: `# vsend <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 `$VAR_ACE`.
 Each Virtual System on the VSX Gateway uses a different `sdopts.rec` file.
 For VS0, copy the file to `/var/ace`.
 On 61000/41000 Security Systems, copy the file to `$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 SmartConsole, 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 VS0, 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+:

1. Configure shared authentication on the Virtual Systems.
 - a) Open SmartConsole.
 - b) From the Gateways & Servers view or Object Explorer, 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 SmartConsole, 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:

1. Configure private authentication on the VSX Gateway and the Virtual Systems.

- a) Open SmartConsole.
- b) From the Gateways & Servers view or Object Explorer double-click the VSX Gateway.
The **General Properties** view opens.

- c) From the navigation tree, select **Other > Legacy Authentication**.
- d) Make sure that **RADIUS** or **TACACS** 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 SmartConsole, install the policy on the Virtual Systems.

Tracking Activity

Use the SmartConsole Logs & Monitor features to show the status of all Security Gateways, VSX Gateways and other virtual devices. To see the status,

The **Overall** status of a virtual device is the most serious status of its Software Blades. For example, if all the **Software Blades** statuses are **OK** except for the SmartEvent blade, which has a **Problem** status, then the **Overall** status will be **Problem**.

Status Icon	Description
 OK	The gateway and all its Software Blades are working properly.
 Attention	At least one Software Blade has a minor issue, but the gateway works.
 Problem	At least one Software Blade reported a malfunction, or an enabled Software Blade is not installed.
 Waiting	SmartView Monitor is waiting for the Security Management Server to send data from Security Gateways.
 Disconnected	Cannot reach the Security Gateway.
 Untrusted	Cannot make Secure Internal Communication between the Security Management Server and the gateway.

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 *Configuring NAT Policy* chapter in the *R80.10 Firewall Administration Guide* <http://downloads.checkpoint.com/dc/download.htm?ID=54806>.

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 these 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 for IPv4, or :: for IPv6.
 - **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, make sure that the VSX Gateway (VS0) can connect to the Internet. Updates are done only through this Virtual System.

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

1. If applicable, configure proxy settings for the VSX Gateway (VS0):
 - a) In SmartConsole, from the **Gateways & Servers** view, double-click the VSX Gateway (VS0).
 - b) From the navigation tree, select **Network Management > Proxy**.
 - c) Configure the proxy settings, and click **OK**.
2. Enable Application Control and URL Filtering on the required Virtual Systems.

Note - You do not have to enable them on the VSX Gateway (VS0).
3. Install policies on the relevant 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, make sure the Software Blade:

- Is enabled and configured on the relevant Virtual Systems and enabled and configured on the VSX Gateway (VS0)
VS0 handles contract validation for all Virtual Systems.
- Can connect to the internet
A Virtual System gets updates through the VSX Gateway (VS0). If the VSX Gateway fails, each Virtual System uses its proxy settings to get the update from the internet.

Note - Where applicable, make sure the routing, DNS, and proxy settings for the VSX Gateway (VS0) are configured correctly.

To enable Anti-Bot and Anti-Virus on Virtual Systems:

1. If applicable, configure proxy settings for the VSX Gateway (VSO) or the Virtual Systems or both:
 - a) From the **Network Object** tree, double-click the VSX Gateway (VSO).
 - b) From the navigation tree, select **Topology > Proxy**.
 - c) Configure the proxy settings, and click **OK**.
2. Enable **Anti-Bot** and **Anti-Virus** on the VSX Gateway (VSO):
 - a) From the **Network Object** tree, double-click the Virtual System.
 - b) In the **Network Security** section, select **Anti-Bot** and **Anti-Virus**.
 - c) Click **OK**.
3. Repeat steps 1-2 for all Virtual Systems that use **Anti-Bot** and **Anti-Virus**.
4. Select the Threat Prevention and configure the policies.
5. Install the Threat Prevention policy (and access policy if needed) on the VSX Gateway (VSO) and the relevant Virtual Systems.

For more about Anti-Bot and Anti-Virus, see the *R80.10 Threat Prevention Administration Guide* <http://downloads.checkpoint.com/dc/download.htm?ID=54828>.

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 R80.10, 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>.

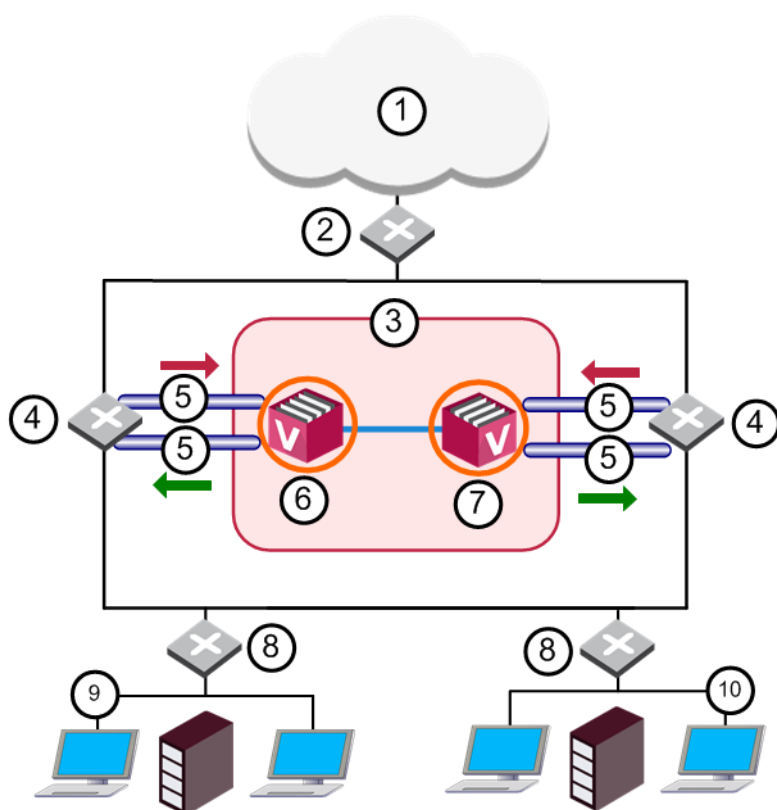
The Trial Period

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.

Virtual System in Bridge Mode

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.



Item	Description
1	Internet
2	Core Network Backbone switch
3	VSX Cluster
4	Router
5	VLAN
6	Member 1
7	Member 2

Item	Description
8	LAN Switches
9	Sales
10	Finance
	Sync Network
	Physical Interface
	VLAN Trunk

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.

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.

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**.

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.

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 **ClusterXL for Bridge Active/Standby**.
4. Reboot the member.

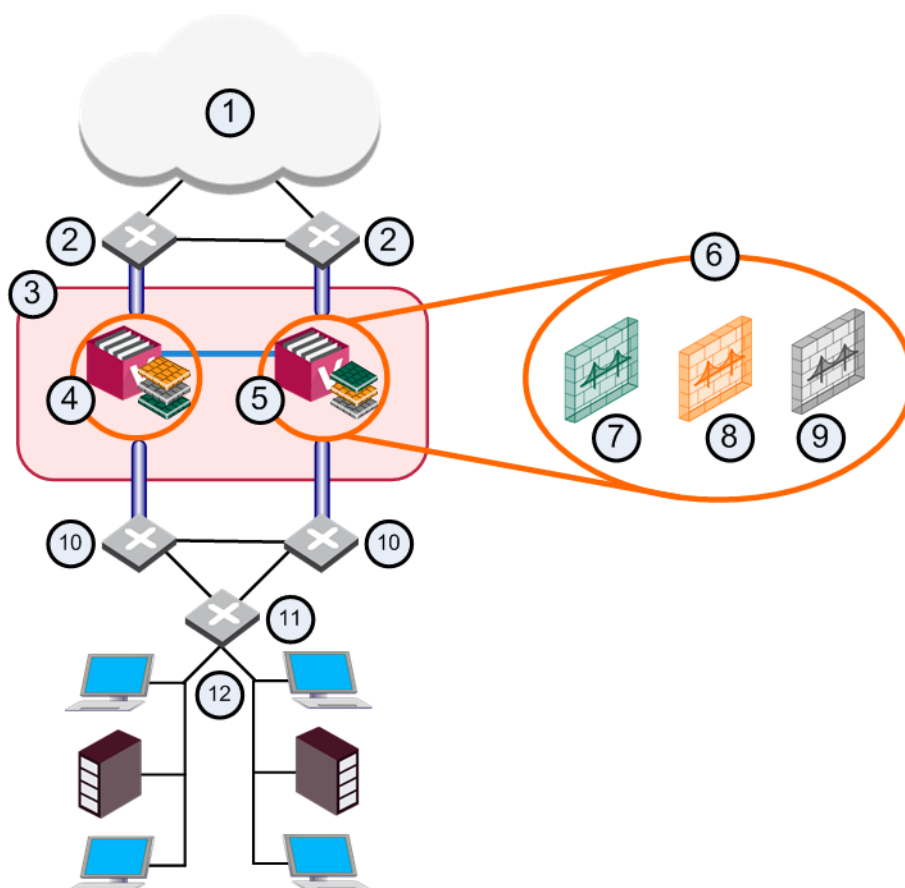
Custom Configuration or Override in Bridge Mode

If you used the **Custom Configuration** template to create the VSX Gateway, or if you selected the **Override Creation Template** option for a Virtual System in Bridge Mode, manually define the network interfaces.

Interfaces: To configure the external and internal interfaces, define interfaces and links to devices in the **Interfaces** table. You can add, change, and remove interfaces. To add an interface, click **Add**. The **Interface Properties** window opens. Select an interface from the list and define its properties.

VLAN Shared Interface Deployment

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



Item	Description
1	Internet
2	Redundant switches (external)
3	VSX Cluster
4	Member 1
5	Member 2
6	Virtual Systems in Bridge Mode
7	VS 1 Active
8	VS 2 Standby

Item	Description
9	VS 3 Backup
10	Redundant switches (internal)
11	VLAN Switch
12	Internal Networks
	Sync Network
	Physical Interface
	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.

VSX Clusters

A VSX cluster has two or more identical, interconnected VSX Gateways for continuous data synchronization and transparent failover. Virtual System Load Sharing (VSL) enhances throughput by distributing Virtual Systems, with their traffic load, among multiple, redundant machines.

Configuring Clusters for Active/Standby Bridge Mode

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

1. Open SmartConsole.
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 Clusters for Active/Active Bridge Mode

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

1. Open SmartConsole.
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 **Standard Layer-2 Loop Detection Protocols**.

Separate Interfaces in Bridge Mode

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

To configure the external and internal interfaces:

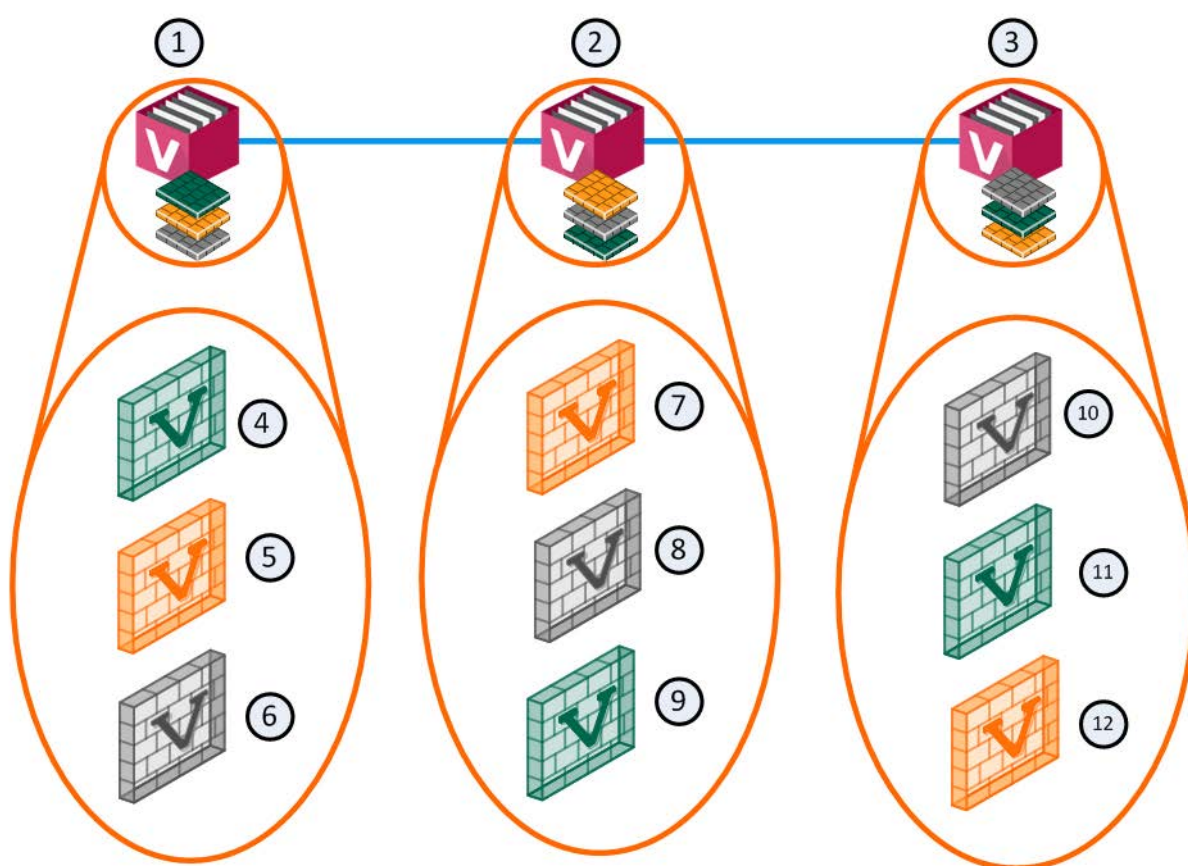
1. Select the desired interfaces for the internal and external networks from the appropriate list.
If the selected Interface is a VLAN interface, enter the same VLAN tag in both the external and internal **VLAN Tag** fields. This field is not available for non-VLAN interfaces.
2. Define the topology for the internal interface:
 - 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. To create a new IP address definition:
 - a) Select **Specific**, and click **New**.
 - b) Select **Group** to define an object group, or **Network** to define network properties.

4. Enable **Layer-3 bridge interface monitoring** to enable layer 3 network fault detection for this Virtual System.
Enter an IP address and subnet mask, which continuously monitors the specified network for faults or connectivity issues. The IP address/subnet define the network on which the Virtual System resides.
5. Complete the definition process ("Completing the Definition" on page 67).

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.



Item	Description
1	Member 1
2	Member 2
3	Member 3
4	VS 1 Active
5	VS 2 Standby
6	VS 3 Backup
7	VS 1 Standby

Item	Description
8	VS 2 Backup
9	VS 3 Active
10	VS 1 Backup
11	VS 2 Active
12	VS 3 Standby
	Sync Network

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.

Converting from High Availability to VSLS

To convert an existing High Availability cluster to VSLS Load Sharing:

1. Close SmartConsole.
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`
6. Enter the Security Management Server or Multi-Domain Security Management Domain Management Server IP address.
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.

Using VSX with Multi-Domain Security Management

In This Section:

Overview	100
VSX Provisioning	101
Defining Multi-Domain Servers	101
Working with Virtual Devices	102

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.

Item	Description
1	SmartConsole
2	Multi-Domain Server
3	Domain Management Server
4	Main Domain Management Server
5	VSX Gateway
6	Virtual Systems in Domain Management Servers

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.

When connected to a Multi-Domain Server, SmartConsole offers 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 SmartConsole, which is accessible only via the Multi-Domain Server, 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 most important difference is that you must first create and configure each Domain and its associated Domain Management Server objects using the SmartConsole connected to a Multi-Domain Server.

Each Domain Management Server is the functional equivalent of one VSX Gateway. You connect to each Domain Management Server with SmartConsole to work with network objects, security policies and other objects for that VSX Gateway.

This is the basic workflow for provisioning a VSX environment in a Multi-Domain Security Management deployment:

1. Define and configure Multi-Domain Server and Multi-Domain Log Server as applicable for your deployment.
2. Create and configure a Domain and Domain Management Server for each VSX Gateway and/or VSX cluster.
3. With SmartConsole, connect to the Domain Management Server to create and configure the VSX Gateway and/or cluster objects ("[Creating a New Cluster](#)" on page 117). Configure the default security policy for these objects as necessary.
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 65) and other virtual devices for each Domain using the SmartConsole of that Domain.

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. See the *R80.10 Multi-Domain Security Management Administration Guide* for conceptual information and detailed procedures for configuring Multi-Domain Servers and Domain Management Servers.

When working with management High Availability, define at least two Multi-Domain Server machines. You can also use multiple Multi-Domain Server machines to efficiently distribute management traffic (management Load Sharing) with more than one Domain Management Server for each Domain. For a load sharing Load Sharing deployment, define a Domain Management Server for each Multi-Domain Server.

Working with Virtual Devices

When working with virtual devices in Multi-Domain Security Management, you must use the applicable Domain Management Server SmartConsole. Otherwise, the configuration procedures are the same to those for a Security Management Server. Multi-Domain Security Management treats virtual devices in the same way 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 a Virtual System to a Domain Management Server

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

1. Launch SmartConsole from the appropriate Domain Management Server.
2. Create and configure the Virtual System ("[Creating a New Virtual System](#)" on page 65).
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. Launch SmartConsole from the appropriate Domain Management Server.
2. Create and configure Virtual Routers ("[Creating a New Virtual Router](#)" on page 74) and Virtual Switches ("[Creating a New Virtual Switch](#)" on page 71) as required.

Introduction to VSX Clusters

In This Section:

VSX Clustering Overview	103
Planning a Cluster Deployment	105
VSX High Availability	106
Virtual System Load Sharing (VSLS)	107
Bridge Mode	113
Using Virtual Switches in a Cluster	115

This chapter presents a conceptual overview of VSX cluster deployments, with emphasis on clustering features and their application. It assumes you are familiar with network cluster applications and environments, particularly ClusterXL.

The Cluster Management chapter ("[Working with VSX Clusters](#)" on page 117) 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.

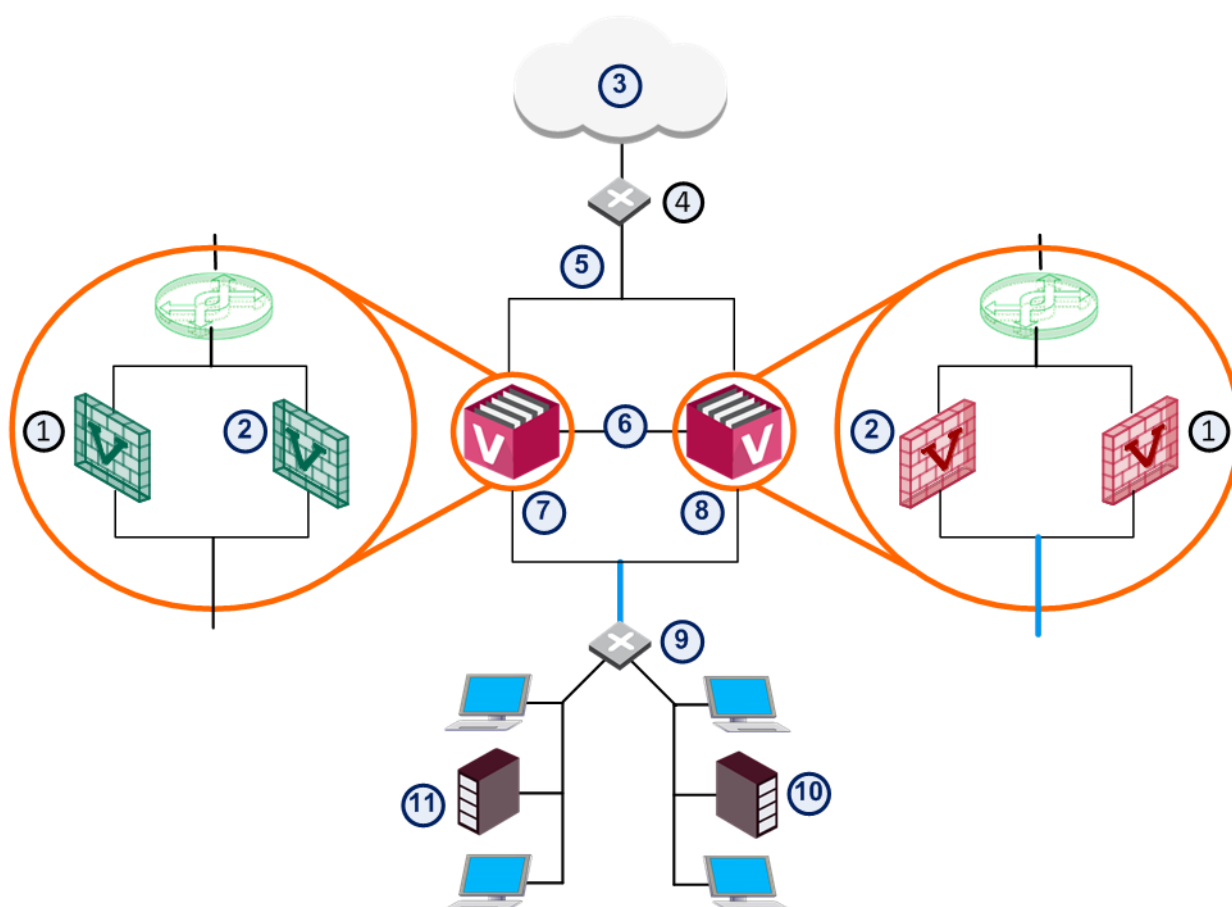
Internal networks send traffic destined for the Internet or external networks, to the cluster IP address. This traffic is processed by the designated cluster member, inspected, and forwarded to its external destination.

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**.



Item	Description
1	VS 2
2	VS 1
3	Internet
4	Router
5	External Cluster Interface
6	Sync

Item	Description
7	Member 1
8	Member 2
9	VLAN switch
10	Network 2
11	Network 1
	VLAN Trunk

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 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:

IPv4 address: 192.168.196.0, netmask: 255.255.252.0 (A range of four class C networks).

IPv6 address and netmask: FD9A::1FFE:0:0:0/80

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:

In 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/standby 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.

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.

Priority	Definition
0	Highest priority, indicating the cluster designated to host the Virtual System active state.
1	Second highest priority, indicating the member designated to host the Virtual System standby state.
> 1	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.

You can change the priority designation ("[Distributing Virtual Systems Amongst Members](#)" on page 134) using the `vsx_util vsls` command.

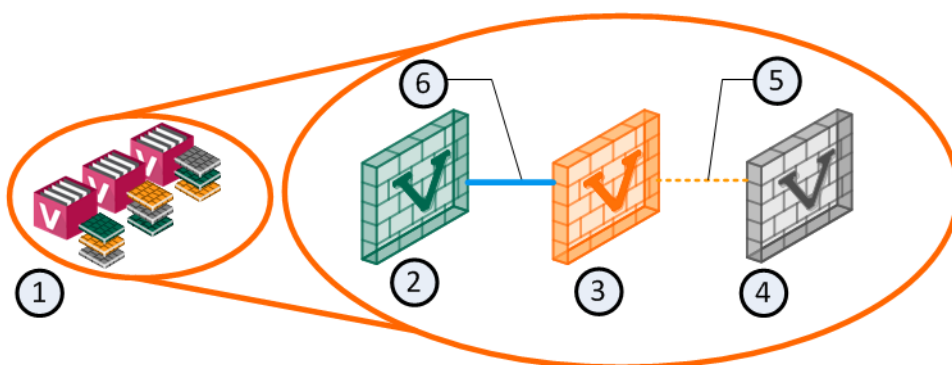
Virtual System Weight

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 134) 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.



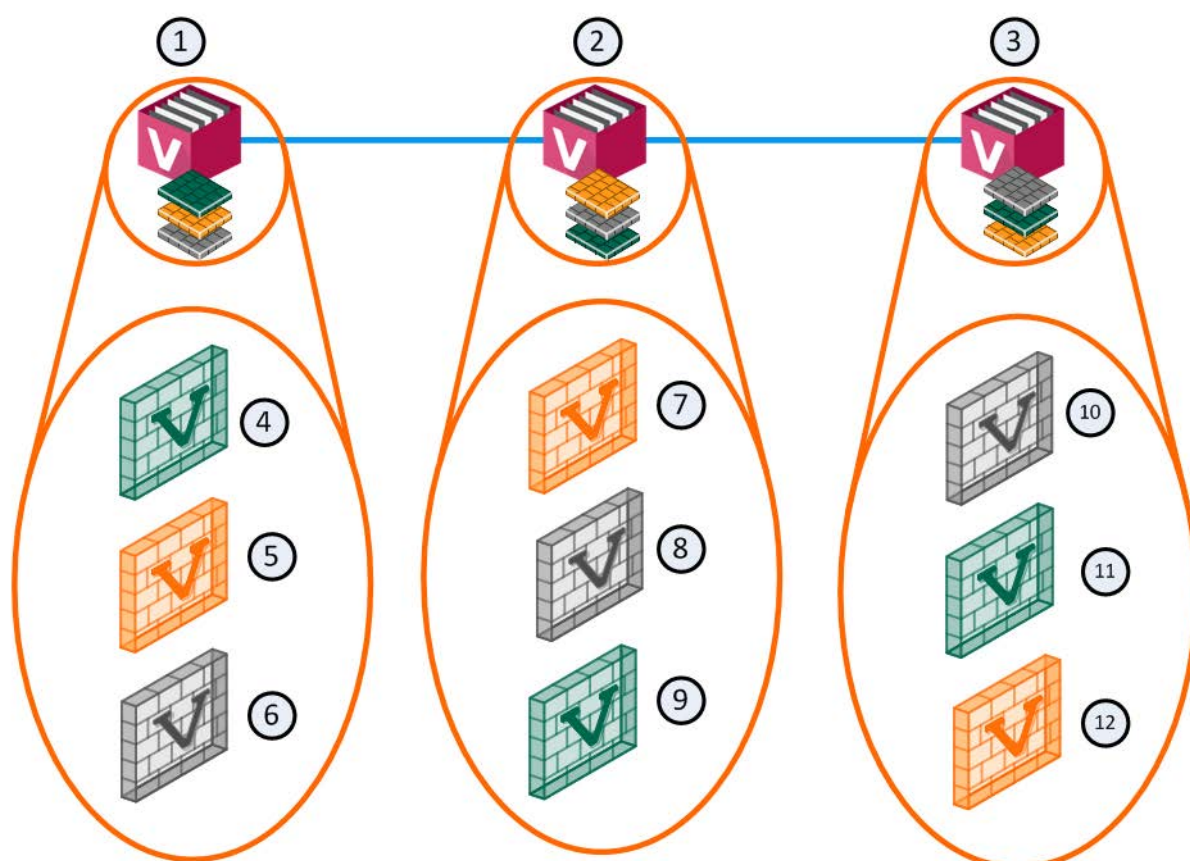
Item	Description
1	VSX Cluster
2	VS1 Member 1 - Active
3	VS1 Member 2 - Standby
4	VS1 Member 3 - Backup
5	Policy updates only
6	State tables synchronized

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.



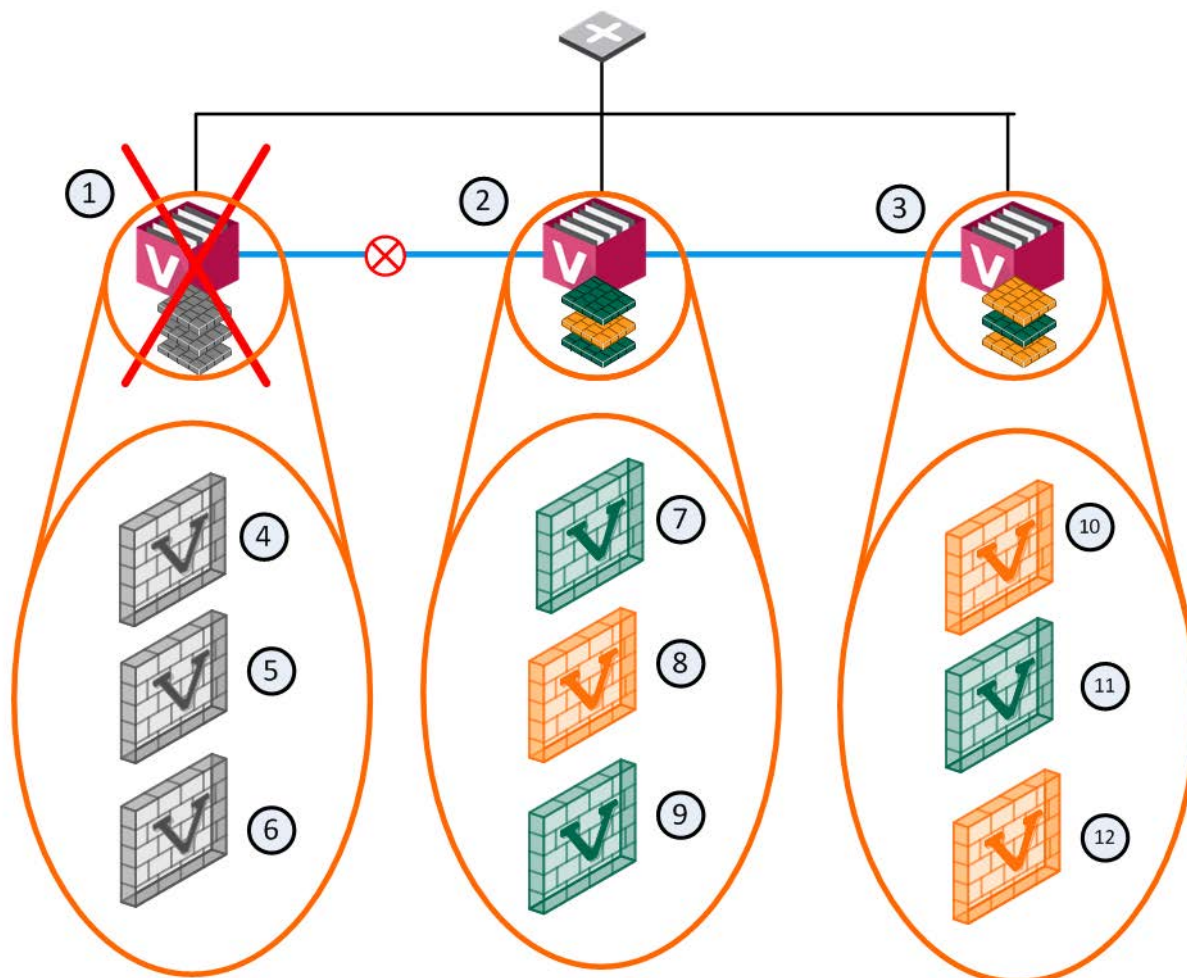
Item	Description
1	Member 1
2	Member 2
3	Member 3
4	VS 1 Active
5	VS 2 Standby
6	VS 3 Backup
7	VS 1 Standby

Item	Description
8	VS 2 Backup
9	VS 3 Active
10	VS 1 Backup
11	VS 2 Active
12	VS 3 Standby
	Sync Network

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, VLS 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.



Item	Description
1	Member 1
2	Member 2
3	Member 3
4	VS 1 Down
5	VS 2 Down
6	VS 3 Down
7	VS1 Active

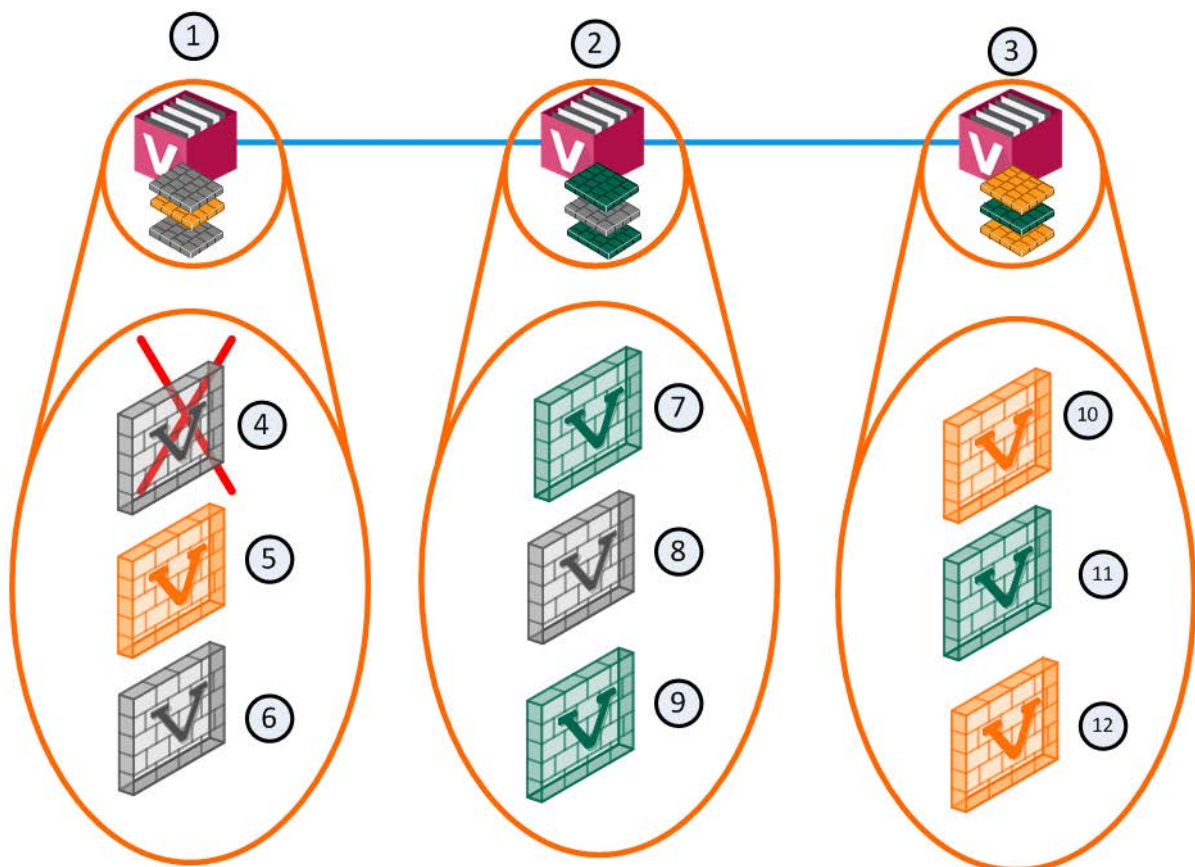
Item	Description
8	VS 2 Standby
9	VS 3 Active
10	VS 1 Standby
11	VS 2 Active
12	VS 3 Standby
	Sync Network
	Network Traffic

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.



Item	Description
1	Member 1
2	Member 2
3	Member 3
4	VS 1 Down
5	VS 2 Standby
6	VS 3 Backup
7	VS 1 Active

Item	Description
8	VS 2 Backup
9	VS 3 Active
10	VS 1 Standby
11	VS 2 Active
12	VS 3 Standby
—	Sync Network

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.



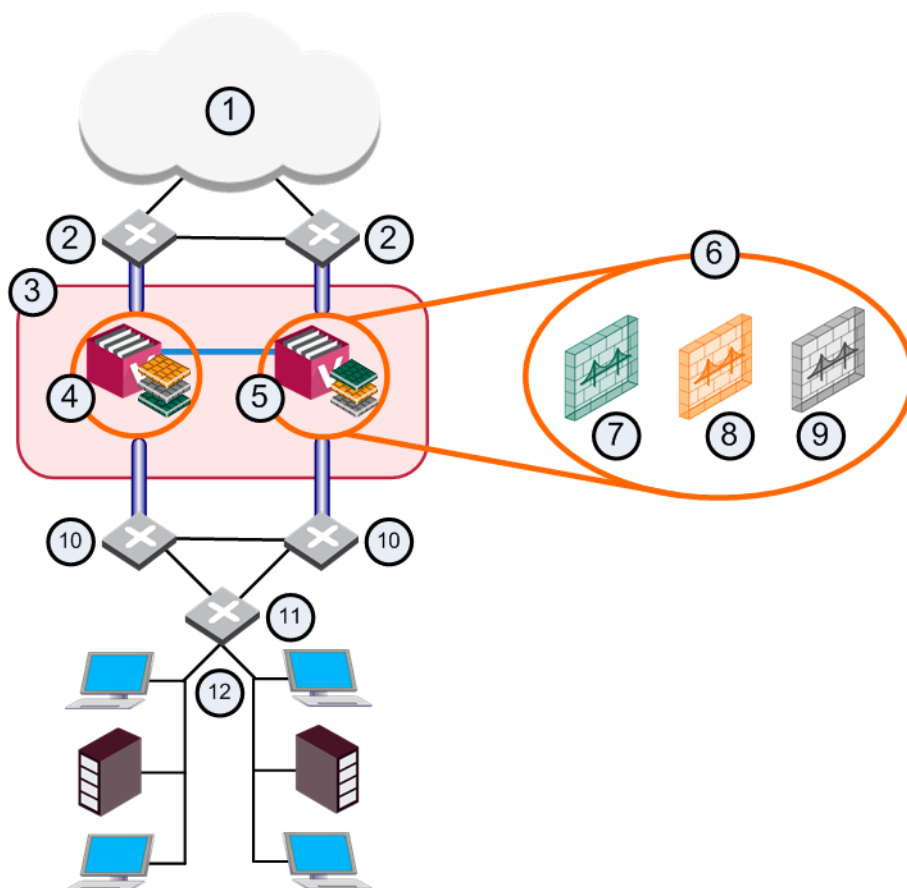
Note - When configuring a Virtual System in the Active/Standby Bridge Mode, you should remove Virtual System VLANs from the STP database in the switches. This action prevents delays due to trunk interface failback.

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 pair of redundant switches through a VLAN trunk. All Virtual Systems in a given member share the same VLAN trunk.



Item	Description	Item	Description
1	Internet	9	VS 3 Backup
2	Redundant switches (external)	10	Redundant switches (internal)
3	VSX Cluster	11	VLAN Switch
4	Member 1	12	Internal Networks
5	Member 2		Sync Network
6	Virtual Systems in Bridge Mode		Physical Interface
7	VS 1 Active		VLAN Trunk
8	VS 2 Standby		

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.

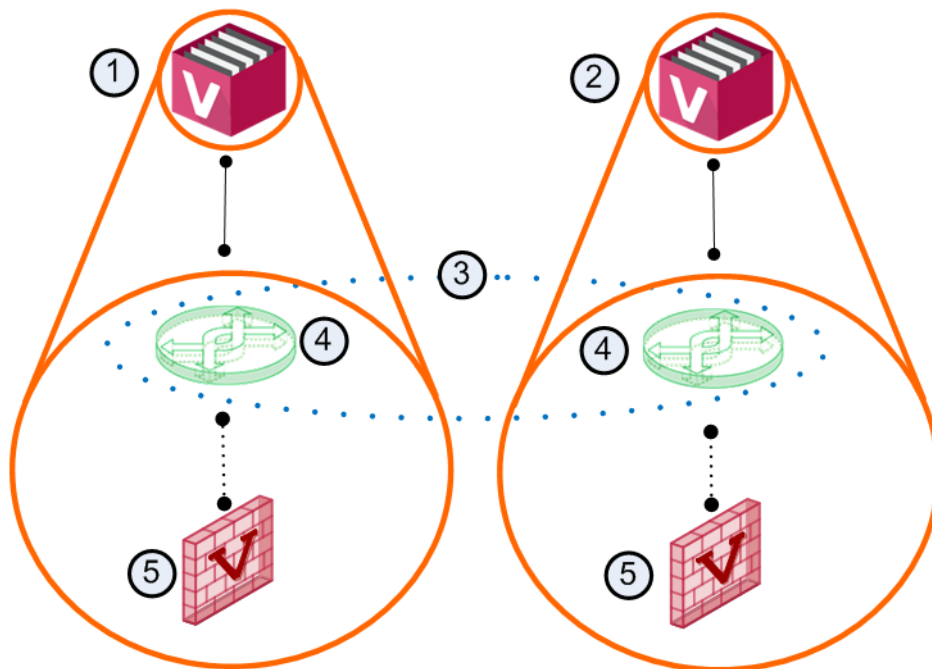
VSX in Active/Standby Bridge Mode is incorporated in the **distribution** layer, enforcing the security policy.

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.



Item	Description
1	Active Member
2	Standby Member
3	Virtual Switch Cluster
4	Active Virtual Switch
5	VS 1
	Physical Link
	Warp Link

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	117
Creating VSX Clusters	117
Modifying a Cluster Definition	120
Working with Cluster Members	125
Changing the Cluster Type	129
Enabling VSX Gateway High Availability	131
Configuring Virtual System Load Sharing	132
Configuring Virtual Systems in Bridge Mode	138
Advanced Clustering Configuration	143

Configuration Overview

You will use SmartConsole for most of the basic cluster configurations. Many cluster management procedures require the command line. For example, you will need the CLI to change members and add Load Sharing definitions.

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 steps to configure a VSX cluster.

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

To create a new cluster:

1. Open SmartConsole.
If you are using Multi-Domain Security Management, open SmartConsole from the Domain Management Server in which you are creating the cluster.
2. From the click **New** and then select **VSX > Cluster**.
The **VSX Cluster Wizard > General Properties** opens.

Defining Cluster General Properties

The **Cluster General Properties** page contains basic identification properties for VSX clusters.

- **VSX Cluster Name:** Unique, alphanumeric name for the cluster. The name cannot contain spaces or special characters except the underscore.
- **VSX Cluster IP Address:** IP address of the cluster. (In R80, use IPv4.)
- **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, **Check Point ClusterXL Virtual System Load Sharing** select 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 IP addresses for the cluster member.

Note: If you define an IPv6 IP address you must also have an IPv4 address.
4. Enter and confirm the activation key to initialize SIC trust between the cluster member and the management server.
5. Follow these steps for all cluster members.
6. Click **Next** to continue.

Defining Cluster Interfaces

The **VSX Cluster Interfaces** window lets you define physical interfaces as VLAN trunks. The list shows all interfaces currently defined on the VSX Gateway or cluster object.

To configure a VLAN trunk:

Select one or more interfaces to define them as VLAN trunks. 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 SmartConsole 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 addresses and net mask for each member.

To use a VLAN as a synchronization interface:

1. Define the VLAN on the Security Gateway.
2. Open SmartConsole 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 *R80.10 Security Management Administration Guide* <http://supportcontent.checkpoint.com/solutions?id=sk111841>.

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. Refer to the troubleshooting steps ("[VSX Diagnostics and Troubleshooting](#)" on page 187) for more information
2. In SmartConsole, double-click the new VSX Cluster object.

Modifying a Cluster Definition

After you create a cluster with the wizard, you can change the topology and other parameters in the **Cluster Members Properties** window. This window lets you configure many advanced features not available with the wizard.

To work with a VSX cluster definition, double-click a cluster object in SmartConsole. The **VSX Cluster Properties** window opens.

You can define most cluster objects with SmartConsole. There are some features or properties that you must CLI commands to configure.

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 applicable product documentation.

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

To manage state synchronization, open the **ClusterXL** window, or run the `vsx_util` command.

- Enable or disable state synchronization.
- Select tracking options for cluster member state changes.

All other ClusterXL configuration properties are disabled.

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

On the **Topology** page, you can see and configure 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 79).
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 79).
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, change, and delete 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+.

Changing the Cluster Management IP and/or Subnet

To add, change or delete the cluster management IP address and/or subnet, run the `vsx_util change_mgmt_ip` ("[change_mgmt_ip](#)" on page 221) and `vsx_util change_mgmt_subnet` ("[change_mgmt_subnet](#)" on page 223) commands.

Changing the Internal Communication Network IP

You can change the internal communication network IP address by using the `vsx_util change_private_net` ("[change_private_net](#)" on page 224) 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 SmartConsole 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 SmartConsole and verify that an object representing the new member appears in the specified cluster.
6. If necessary, modify the cluster configuration.
7. Close SmartConsole.
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 SmartConsole.
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) Select the cluster name.
 - e) Select 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 SmartConsole, the object for the deleted member no longer appears in the specified cluster.
5. Open SmartConsole 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.

To upgrade the cluster and its members:

1. Close SmartConsole.
2. From the management server command line, enter Expert mode.
3. Run: `vsx_util upgrade`
Enter when prompted:
 - Security Gateway or main Domain Management Server IP address
 - Administrator name and password
 - Cluster name
 - Select the version to upgrade to
4. Wait until the `Finished upgrading/database saved successfully` message appears, indicating that the database has been updated and saved.
5. Open SmartConsole 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.
6. Install VSX on each upgraded member.

7. Run the initial configuration steps on each member:
 - 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.

If you upgrade with CPUSE, you can skip the next step and reboot when the gateway is ready.

8. Run: `vsx_util reconfigure`

Enter 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.

9. Wait until the `Finished upgrading/database saved successfully` message appears.
10. 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 SmartConsole.
2. From the CLI, enter Expert mode.
3. Run `vsx_util vsls`
4. Enter the administrator user name and password.
5. Enter the IP address for the Security Management Server or Domain Management Server.
6. Select the VSX cluster name.
7. From the Load Sharing menu, enter **3. Set all VSs active on one member**.
8. Enter the number of the cluster member that is hosting the 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 HA:

1. Run `vsx_util convert_cluster`
2. Enter the IP address for the Security Management Server or Domain Management Server.
3. Enter the administrator user name and password.
4. From the Load Sharing menu, enter **3. Set all VSs active on one member**.
5. Enter or select the VSX cluster name.
6. Enter **HA**.
7. Reboot all members.

Converting from High Availability to VSLS

To convert an existing High Availability cluster to VSLS Load Sharing:

1. Close SmartConsole.
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`
6. Enter the Security Management Server or Multi-Domain Security Management Domain Management Server IP address.
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.

This sample output is for versions R77.10 and higher.

```
vsx_util convert_cluster

*****
Note: the operation you are about to perform changes the information in the
management database. Back up the database before continuing.
*****

Enter Security Management Server/main Domain Management Server IP address
(Hit 'ENTER' for 'localhost'):
Enter Administrator Name:
Enter Administrator Password:
Select VSX cluster object name:
  1)    vsx_cluster_A
  2)    vsx_cluster_B
Select: 1
Enter desired ClusterXL mode: HA-High Availability, LS-Load Sharing
(HA | LS): LS

All modules must be in the 'Per VS aState' mode to conclude this operation
successfully. Use the command cpconfig on each module to verify compliance
before continuing with the operation.
When converting a cluster, there are two options for distributing the
existing Virtual System(s) among cluster members:
1. Distribute all Virtual Systems so that each cluster member is equally loaded.
2. Set all Virtual Systems as Active on the same cluster member.

After converting the cluster, the command vsx_util redistribute_vsls may
be used to modify Virtual System distribution.
Enter distribution option (1-2) [1]: 1

Converting the cluster to ClusterXL Load Sharing mode...
The cluster was successfully converted to ClusterXL Load Sharing mode
Installing new policy...
...
Policy installation finished successfully.
```

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 Security Management Server, enter Expert mode.
2. Run `vsx_util vsls`
3. Enter the IP address for the Security Management Server.
4. Enter administrator user name and password.
5. Select the VSX Cluster name.
6. From the VSLS menu, select the configuration option.

Enabling VSLS

In order to use VSLS 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 VSLS. 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 VSLS Cluster

To create a new VSLS cluster:

1. Open SmartConsole.
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 117).
 - 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 118).
 - 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 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
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_util vsls` command:

1. Run `vsx_util vsls` from the Expert mode on the management server
2. Select the desired choice from the **VSLs** menu

vsls_config vsls main menu

This sample output is for versions R77.10 and higher.

```

Enter Administrator Name: aa
Enter Administrator Password:
Select VSX cluster object name:
  1)    vsx_cluster_A
  2)    vsx_cluster_B
Select: 1
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

Methods to change priorities and weights:

- 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 priorities and weights to individual Virtual Systems.

Note: After you save changes, the update requires time (several minutes or longer), depending on the quantity of Virtual Systems and cluster members.

To automatically assign weights to all Virtual Systems:

1. From the VSLs menu, select **Manually set priority and weight**.
2. Scroll through each Virtual System. Enter: a

- For each Virtual System, enter a weight value and press **Enter**.

If you do not enter a weight value for a Virtual System, the assigned weight is not changed.
Only Virtual Systems with a new weight value are updated.

To stop entering weight values, enter s.

- At the **Save & apply configuration** prompt, enter: y

To manually update both priorities and weights for individual Virtual Systems:

- From the VSLS menu, select **Manually set priority and weight**.
- Enter: m
- At the **Would you like to change the Virtual System's priority list?** prompt, enter: y
- Enter the number of the member to get the highest priority.
- Enter the number of the member to get the next highest priority.
- Continue until all members have a priority.
- At the **Would you like to change the Virtual System's weight?** prompt, enter: y
- Enter the new weight value. A valid value is an integer between 1 and 100.
- At the **Do you wish to configure another Virtual System?** prompt, enter y to configure a different Virtual System or n to continue.
- At the **Save & apply configuration?** prompt, enter: y

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"

VSID	VS name	gw150	gw151	gw152	Weight
2	vs1	0	1	2	10
3	vs2	2	0	1	10
4	vs3	1	2	0	10
5	vs5	0	2	1	10
6	vs4	1	0	2	10
Total weight		20	20	10	50

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.

Priority	Definition
0	Highest priority, indicating the member designated to host the Virtual System active state.
1	Second highest priority, indicating the member designated to host the Virtual System standby state.
> 1	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.

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

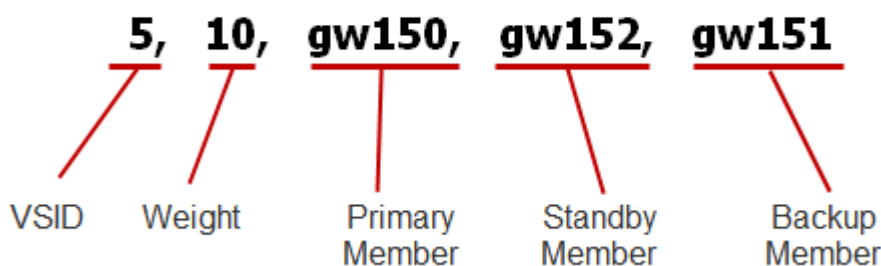
# Virtual System name: vs2
3,10,gw151,gw152,gw150

# 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:



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 SmartConsole.

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.

Command	Action
!comments	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.
!verbose	Displays whether or not each data line has been successfully verified and the configuration parameters for each Virtual System.
!log	Saves !comments and !verbose information in the vsx_util.log file.

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 **ClusterXL 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 SmartConsole.
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 **Standard Layer-2 Loop Detection Protocols**.

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 139).

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 SmartConsole.
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**.

Multi Bridges

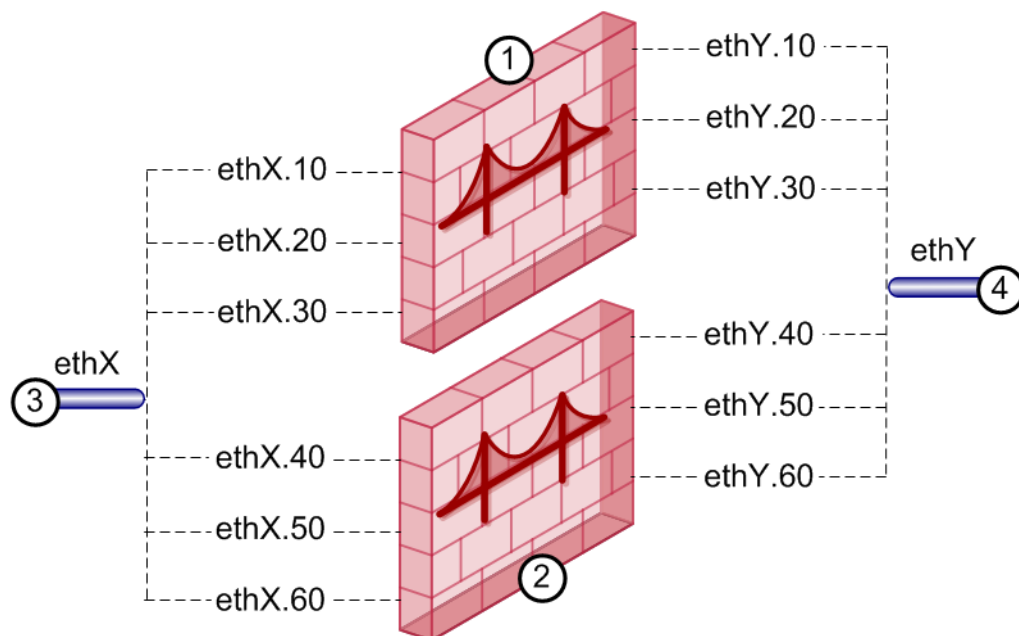
This feature is supported only in R77.30 and higher, for VSX Gateways, and VSX clusters in Active/Active Bridge mode.

Multi Bridge allows traffic from many different VLANs to move over one Virtual System in Bridge mode. In a Virtual System in Bridge mode, you can add physical and VLAN interfaces. When you add more than two VLAN interfaces, Multi Bridge is automatically enabled. Configure the same VLAN tag on each set of two interfaces to make them bridged.

Requirements for Multi Bridge interfaces:

- All interfaces must be VLANs.
- You can make multiple bridges only between two VLAN trunks.
- You can add up to 64 pairs of VLAN interfaces for one Multi-bridge.
- Those two VLAN trunks must be used together, and not with other VLAN trunks, in other Virtual Systems in Bridge mode or Multi Bridges.

For example, you define eth1.10, eth2.10, eth1.20, eth2.20. Now the VLAN trunks, eth1 and eth2, cannot be used with other VLAN trunks on other Virtual Systems in Bridge mode: eth1.30 cannot bridge with eth3.30.



Item	Description
1	Virtual System in Bridge Mode with two bridges on VLAN interfaces 81 and 82.
2	Virtual System in Bridge Mode with three bridges on VLAN interfaces 40, 50, and 60.
3 and 4	VLAN Trunks. Each must be paired with the other in all bridges, or used without bridging. They cannot be paired with a different trunk.

To define a new Multi Bridge:

1. In SmartConsole, right-click the VSX gateway and select **Checkpoint > VSX**.
2. Click **Virtual System**.
3. Enter a name for the bridge.
4. Select **Bridge Mode**.
5. Click **Next**.
6. Click **Add** and add a VLAN interface for the bridge.
7. Click **Add** again to add the second VLAN interface.
8. Add more VLAN interfaces to the multi bridge in the same way.
Make sure the interfaces in each pair have the same VLAN tag, from different interfaces.
For example: eth2.50, eth2.51, eth3.50, eth3.51
Make sure you keep using the same two VLAN trunks.
9. Click **Next**.
10. Click **Finish**.
11. Install policy.

To convert a bridge to a Multi Bridge:

1. In SmartConsole, double-click the Virtual System in Bridge mode.
2. In the **Bridge Configuration** window, click **Topology**.
If there are physical interfaces in the **Interfaces** list, delete them.
3. Click **Add**, to add more VLAN interface pairs.
4. Click **OK**.
5. Install policy.

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:

Default Value Of Fifth Byte	Purpose
0xfe	CCP traffic
0xfd	Forwarding layer traffic

- 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:

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

Parameter	Default value
fwha_mac_magic	0xfe
fwha_mac_forward_magic	0xfd

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`.

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.

Configuring CoreXL in a VSX VSLS Cluster

In VSX VSLS Clusters of version R80.10 and higher, changing the CoreXL configuration or running the `cphastop` command on VS0 results in failover of all the active virtual systems on that member. This behavior is by design. Active virtual systems on other members do not fail over.

Working with Link Aggregation

In This Section:

Link Aggregation Overview	146
Configuring High Availability Mode.....	150
Configuring Load Sharing Mode	153
Configuring Cisco Switches for Load Sharing.....	155
Troubleshooting Bonded Interfaces	156
Gaia CLI Reference	157

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.

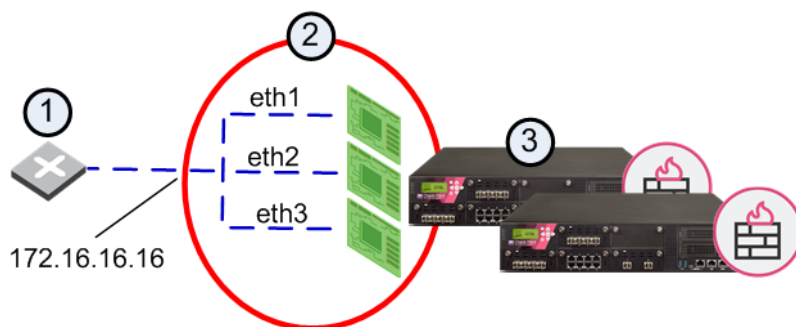
For more about Link Aggregation, see the *R80.10 ClusterXL Administration Guide* <http://downloads.checkpoint.com/dc/download.htm?ID=54804>.

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.



Item	Description
1	Switch
2	bond 0
3	Cluster

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

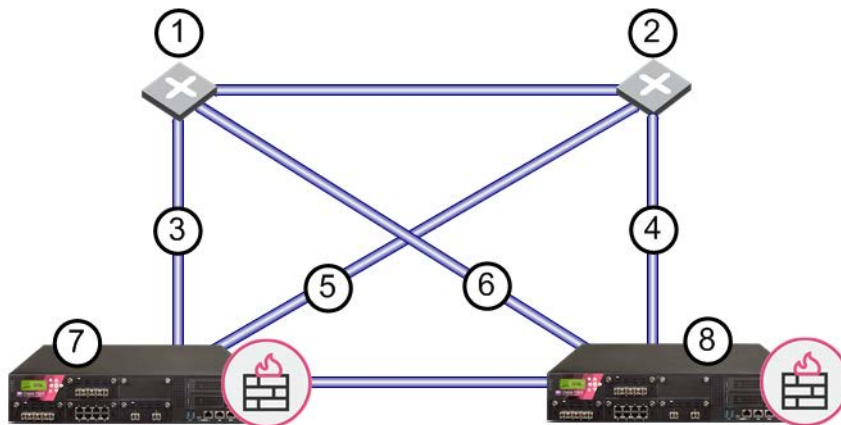
Clusters, by definition, provide redundancy and high availability at the gateway level. Link Aggregation, however, adds interface and switch redundancy by providing automatic failover to a standby interface card within the same VSX Gateway.

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.



Item	Description
1	S-1
2	S-2
3	C-1
4	C-4

Item	Description
5	C-2
6	C-3
7	Member 1
8	Member 2

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 154), 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

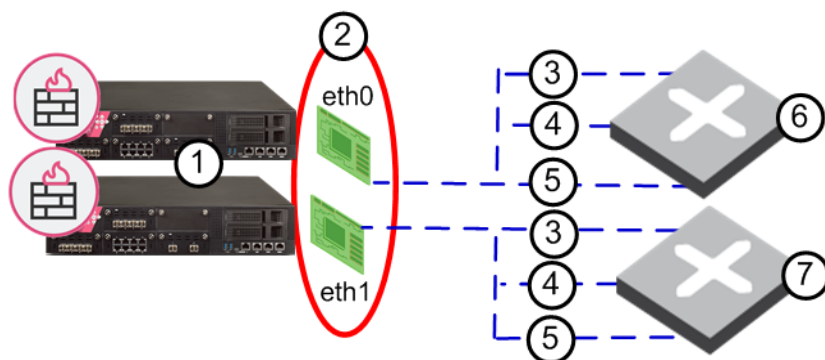
CCP failover occurs only when other cluster members are not down, in this sequence.

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.



Item	Description
1	Cluster
2	bond 0
3	VLAN 1
4	VLAN 2

Item	Description
5	VLAN 3
6	S-1
7	S-2

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 SmartConsole 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 151).

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 SmartConsole. 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 - Gaia

To configure the newly created bond for a Gaia cluster:

1. Open SmartConsole.
2. Delete the slave interfaces from the bond that you are not using.
 - a) From the navigation tree, click **Topology**.
 - b) From the navigation tree, click **Physical Interfaces**.
 - c) Select the slave interface, and click **Remove**.
 - d) Click **OK**.
 - e) Do these steps again for all the slave interfaces.
3. From clish, create the new bond interface ("[Creating or Deleting a Bond Interface](#)" on page 158).

4. Open SmartConsole and from the **Network Objects** tree, double-click the VSX Gateway or cluster object.
5. From the navigation tree, click **Physical Interfaces**.
6. Click **Add**, and configure the bond interface.
The **Physical Interface Properties** window opens.
 - a) Enter the bond name.
 - b) If the bond is a VLAN trunk, select **VLAN Trunk**.
 - c) Click **OK**.
7. From the navigation tree, click **Topology**.
8. 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**.
9. Install the policy.

Reconfiguring the Bond - SecurePlatform

To configure the newly created bond for a SecurePlatform cluster:

1. Open SmartConsole.
2. From the **Object Explorer**, 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.
 - b) If the bond is a VLAN trunk, select **VLAN Trunk**.
 - c) 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 vxutil change_interfaces](#)" on page 153).

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.

1. Close SmartConsole for all Multi-Domain Security Management Domain Management Servers using the designated interface.
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 154).

4. For configurations that use Performance Pack, configure the core affinities ("[Setting Affinities](#)" on page [154](#)).
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 SmartConsole 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 [151](#)).

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:

```
$FWDIR/conf/cpha_bond_ls_config.conf
```

Each line of the file should be of the following syntax:

```
<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.



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:
`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:

bond0	bond1
eth0	eth3
eth1	eth4
eth2	eth5

Two of the cores will need to handle two interfaces each. An optimal configuration can be:

bond0		bond1	
eth0	core 0	eth3	core 0
eth1	core 1	eth4	core 1
eth2	core 2		
		eth5	core 3

Configuring Cisco Switches for Load Sharing

These are sample configuration commands for Cisco switches.

For 802.3ad:

```
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:

```
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

Troubleshooting Workflow

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 *R80.10 ClusterXL Administration Guide*
<http://downloads.checkpoint.com/dc/download.htm?ID=54804>.
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 is a quick reference for Link Aggregation commands. 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

Parameter	Description
bondID	ID of bond, an integer between 1 and 1024
IFName	Name of interface to add to the bond
primary	Name of primary interface in the bond
mii-interval	Frequency that the system polls the Media Independent Interface (MII) to get status
up-delay down-delay	Waiting time to confirm the interface status before taking the specified action (0-5000 ms, default = 200 ms)
mode	Bond operating mode (" Defining the Bond Operating Mode " on page 158)

Parameter	Description
lacp-rate	Link Aggregation Control Protocol packet transmission rate: - slow - LACPDU packet sent every 30 seconds - fast - LACPDU packet sent every second
xmit-hash-policy	Algorithm for interface selected by TCP/IP layer

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** (users 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.....	162
Monitoring Memory Resources.....	170
VSX CPU Monitoring Commands	173
SNMP Monitoring.....	176
SNMP Monitoring of VSX	178
Configuring Jumbo Frames	185

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.

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 done for each interface, but for all the system. One class of services applies to all traffic entering the VSX Gateway or cluster, regardless of the specified interface from which the traffic originates.



Note - On multiple-CPU machines, enforcement is not done system-wide, but for each CPU. Global enforcement is done separately on traffic processed by each CPU.

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` ("[The cpqos Command](#)" on [page 165](#)) 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

```
cpqos - Manage the network quality of service.

cpqos install          - Install the QoS policy.
cpqos uninstall        - Uninstall the QoS policy.
cpqos status           - Check if policy is installed.
cpqos class show [-b]  - Show the QoS policy. [-b] display dscp in binary numbers.
cpqos class add <name> prio <val> type <llq|reg> [weight <val>] dscp
<val[,val2[,val3...]]>
                        - Add new class with specified name
                        prio   - priority (1-15)
                        type   - low-latency or regular
                        DSCP   - values "default" or 0-63 (decimal or 8 bits
binary)
                        weight - (1-1000) for classes of type "reg"
cpqos class del <name> - Delete specified class name.
cpqos stats [-u]       - Show statistics. [-u] will show statistics per CPU.
```

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 168)

cpqos class add

This command adds a class with the following arguments:

Argument	Value
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.
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:

Column	Value
rx	Number of bytes that arrived to this class since the last time statistics were presented
tx	Number of bytes that were transmitted from this class since the last time statistics were presented
drops	Number of bytes that were dropped from this class since the last time statistics were presented

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

Traffic Type	Meaning...
Diamond	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).
Platinum	Real-time traffic with low bandwidth requirements that is less sensitive to latency and drops than Diamond.
Gold	Traffic which is sensitive to drops
Silver	Traffic which is less sensitive to drops than Gold.
Bronze	Various types of traffic which require resource allocation. This traffic is usually assigned to the Best-Effort PHB.
Copper	High-volume traffic with a tendency to consume bandwidth

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 than 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:

```
cpqos class add Diamond type llq prio 1 dscp 46
cpqos class add Platinum type llq 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: llq
    weight: 0
    DSCPs: 46

class: Platinum
    priority: 2
    type: llq
    weight: 0
    DSCPs: 32

class: Gold
    priority: 3
    type: reg
    weight: 100
    DSCPs: 26

class: Silver
    priority: 4
    type: llq
    weight: 100
    DSCPs: 28

class: Bronze
    priority: 5
    type: llq
    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:

class	priority	type	weight	rx	tx	drops
Diamond	1	llq	0	2775	2650	0
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}
```

Parameter	Description
enable	Enables memory resource monitoring. Reboot the VSX Gateway.
disable	Disables memory resource monitoring.
status	Shows if memory resource monitoring is enabled or disabled.

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>]
```

Parameter	Description
<code>-vs</code>	Shows the memory usage of the specified virtual devices.
<code><VSID></code>	The ID of the virtual device. To show multiple devices: • Put a space between each VSID: <code>-vs 1 3 5</code> • List a range of VSIDs: <code>-vs 1-4</code> Note - You can combine VSID ranges together with single VSIDs
<code>unit</code>	Change the memory measurement unit shown in the command output.
<code><measure></code>	The memory measurement unit. The default value is megabytes. The values are: • B - bytes • K, KB - kilobytes • M, MB - megabytes (default) • G, GB - gigabytes
<code>sort</code>	Sort the results according to the virtual devices that use the most memory.
<code><top></code>	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.

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)

```
VSX Memory Status
=====
Memory Total: 997.22 MB
Memory Free: 232.56 MB
Swap Total: 2047.34 MB
Swap Free: 2047.16 MB
Swap-in rate: 0.00 MB
VSID | Memory Consumption
=====+=====
  0 |           133.50 MB
  8 |           92.41 MB
  3 |           43.81 MB
  6 |           42.47 MB
  1 |           42.47 MB
```

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>
```

Parameter	Description
swap	Configures the swap-in sample rate for the system.
<minutes>	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.

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
```

VSID	Private_Clean	Private_Dirty	DispatcherGConn
0	13544.00 KB	144268.00 KB	0.00 KB
1	1740.00 KB	46276.00 KB	0.00 KB
2	1720.00 KB	46868.00 KB	0.00 KB
3	1720.00 KB	46644.00 KB	0.00 KB
4	1712.00 KB	45144.00 KB	0.00 KB
5	1712.00 KB	45836.00 KB	0.00 KB
6	1720.00 KB	45000.00 KB	0.00 KB
7	1720.00 KB	45044.00 KB	0.00 KB

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.



Note - You must enable VSX Resource Control Monitoring to see data about CPU usage for each Virtual System over SNMP.

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}
```

Parameter	Description
enable	Enables Resource Monitor
disable	Disables Resource Monitor
show	Displays whether Resource Monitor is enabled or disabled

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
```

Parameter	Description
-u	Displays information per CPU (SMP only).
-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
ID Name                               |1sec  10sec 1min  1hr   24hr*
=====+=====
0 VSX2                                |0.11  0.06  0.08  0.07  0.01
1 VSX2_vs1                            |15.80 21.57 21.75 22.28 1.94
2 VSX2_vsw                            |0.00  0.00  0.00  0.00  0.00
3 VSX2_vs2                            |16.91 22.57 22.77 23.09 2.01
=====+=====
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
```

ID	Name	CPU	1sec	10sec	1min	1hr*	24hr*
0	vsxb	0	0.30	0.44	0.48	0.00	0.00
		1	12.50	1.33	0.48	0.00	0.00
		2	3.40	1.36	5.25	0.00	0.00
		3	0.20	0.24	0.41	0.00	0.00
		Avg.	4.10	0.84	1.65	0.00	0.00
1	vs1	0	0.00	0.02	0.11	0.00	0.00
		1	0.00	0.01	0.01	0.00	0.00
		2	0.20	0.04	0.11	0.00	0.00
		3	0.00	0.01	0.01	0.00	0.00
		Avg.	0.05	0.02	0.06	0.00	0.00
Total Virtual Devices CPU Use		0	0.30	0.46	0.59	0.00	0.00
		1	12.50	1.34	0.48	0.00	0.00
		2	3.60	1.40	5.35	0.00	0.00
		3	0.20	0.25	0.42	0.00	0.00
		Avg.	4.15	0.86	1.71	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

For more about using SNMP, see *SNMP* in the *R80.10 Gaia Administration Guide*
<http://downloads.checkpoint.com/dc/download.htm?ID=54824>.

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
ifDescr
```

Sample SNMP VS Mode Configuration

Sample remote query of interfaces for VS2:

```
> snmpwalk -n ctxname_vsid2 -v 3 -l authNoPriv -u admin -A abcd1234 <VSX_IP_Address>
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 ("[VSX CPU Monitoring Commands](#)" on page [173](#)) to see information about CPU usage per Virtual System.

SNMP Monitoring of VSX

In This Section:

VSX SNMP Modes	178
Supported SNMP Versions	181
Configuring SNMP VSX modes	181
The VSX SNMP Tree	184

VSX SNMP Modes

VSX supports these SNMP modes:

- SNMP Default Mode
- SNMP VS Mode
- SNMP VS in vs-direct access mode

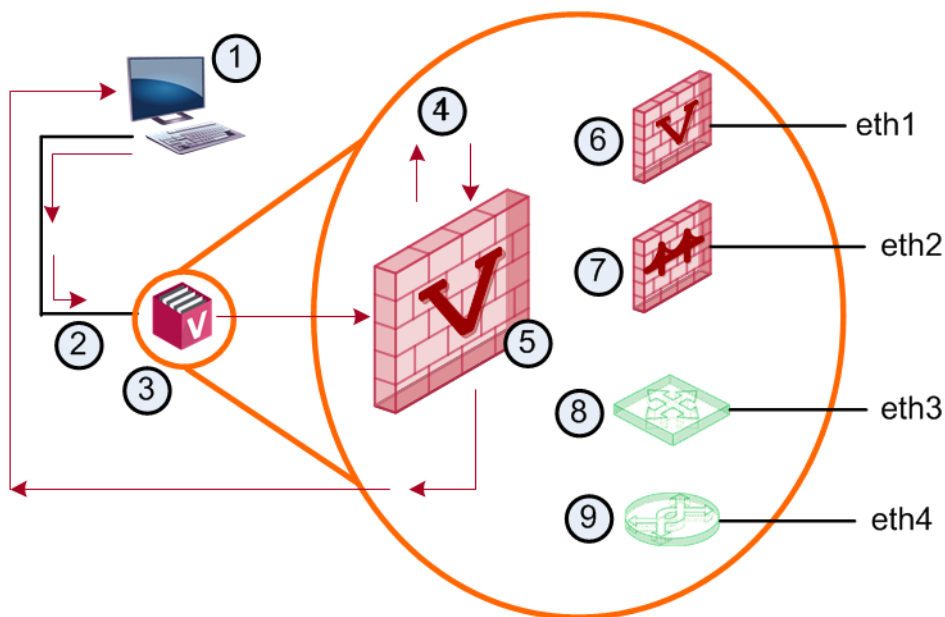
For SNMP Best Practices, see the Best Practices Guide

http://supportcontent.checkpoint.com/documentation_download?ID=31396.

SNMP Default Mode

In SNMP default mode:

- The SNMP daemon runs only in the context of VS 0 (the VSX gateway).
- The VS0 SNMP daemon has a set of tables with counters (VSX SNMP tree) for each Virtual Device.
- SNMP queries must be sent to IP address of the VSX gateway (context is VS 0).



Item	Description
1	Query Host
2	eth0
3	VSX Gateway
4	SNMP Daemon
5	VS 0

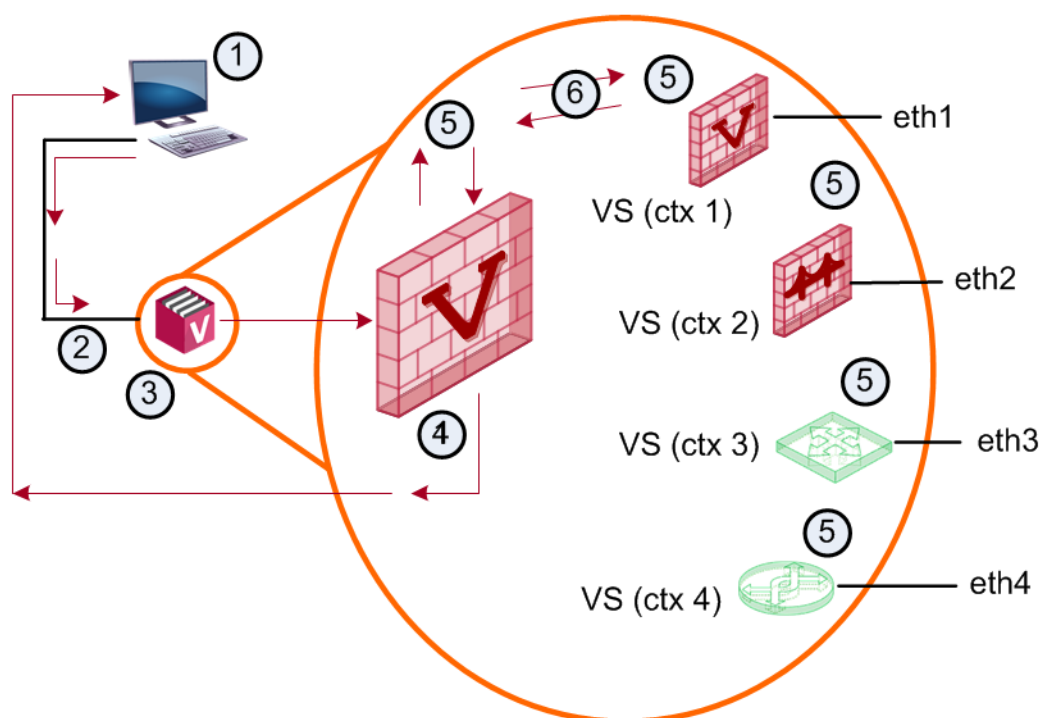
Item	Description
6	VS (ctx 1)
7	VSB (ctx 2)
8	VR (ctx 3)
9	VSW (ctx 4)

SNMP VS Mode

In SNMP VS mode:

- Each Virtual Device has separate SNMP daemon running in the context of that Virtual Device.
- Query for virtual devices uses the VS 0 IP address.
- You must run the SNMP query using the interface on the VSX gateway.
 - The query is relayed to the specified virtual device.
 - The virtual device sends the response through the same VSX Gateway interface.
- The VS ID must be specified in the SNMP query.

Note - Default mode query functionality is not decreased when you enable SNMP VS mode.



Item	Description
1	Query Host
2	eth0
3	VSX Gateway

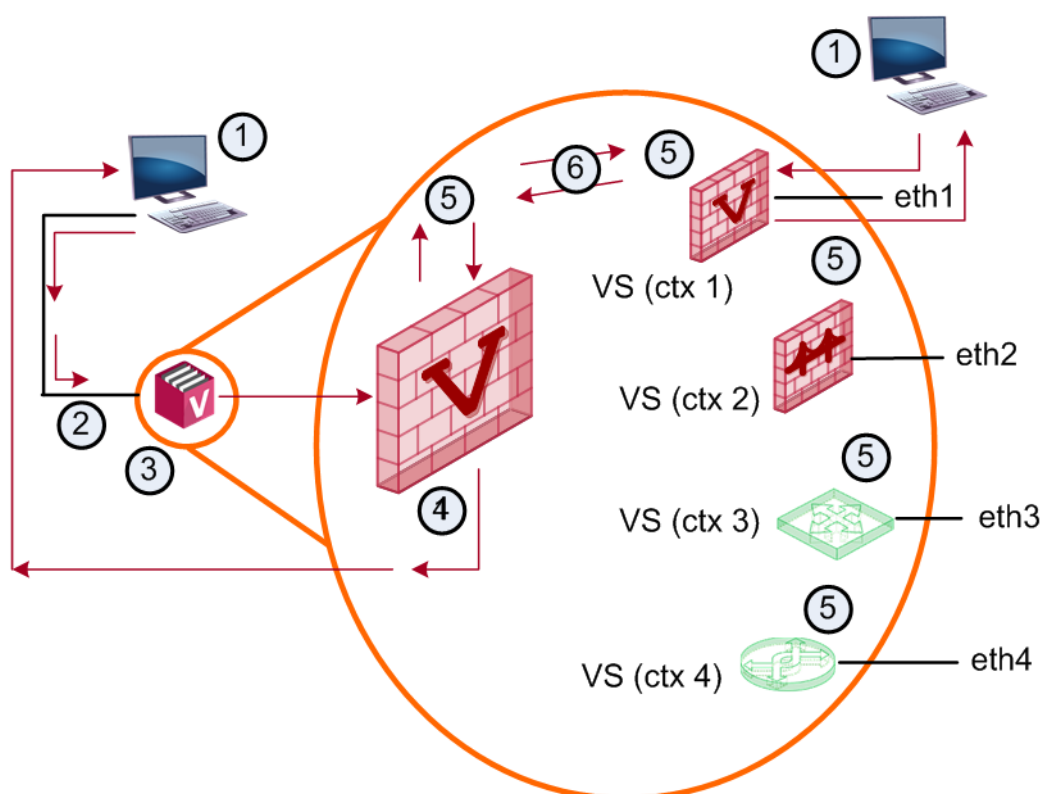
Item	Description
4	VS 0
5	SNMP Daemon
6	UDS

SNMP VS in vs-direct-access mode

- Enables SNMP queries on the IP address of a virtual device (not only VS 0).

Note -

- For cluster deployments, you can query only virtual IP addresses.
- Only virtual devices with an IP address can be queried, not virtual switches or bridges.
- SNMP VS in vs-direct-access mode is available only when SNMP VS mode is enabled.



Item	Description
1	Query Host
2	eth0
3	VSX Gateway

Item	Description
4	VS 0
5	SNMP Daemon
6	UDS

Supported SNMP Versions

SNMP V1, V2c, and V3 are supported in all monitor modes.

Note - For SNMP queries of virtual devices using the VS 0 IP address:

- **SNMP V1 and V2c**
Query the virtual device using the VSID and the configured SNMP community.
- **SNMP V3**
Query the virtual device using the VSID and SNMP V3 context mechanism.

Configuring SNMP VSX modes

Each Virtual System must meet these requirements:

SNMP USM user

- To use SNMP V3 queries, an SNMP USM user must be defined. For more on USM user creation commands, see the *R80.10 Gaia Administration guide*.
- To use SNMP V3 queries on VSX, the USM user must be configured with the allowed Virtual-Devices:
`set snmp usm user <user_name> vsid <VSID>`
- By default, a USM user in VSX has no allowed Virtual-Devices.

Allowed interfaces

If you enable *vs-direct-access mode*, the Virtual System accepts SNMP queries on all the interfaces. To prevent SNMP queries for a specified interface, add a new rule to the policy that blocks SNMP traffic on that interface.

Query source

In *vs mode* and *vs-direct-access mode*, there is no specification for query source. All sources allowed in the Security Policy are valid.

Running SNMP Queries

When you query a Virtual System Load Sharing cluster with the VSX Gateway (VS 0) Virtual IP address, the Virtual System on the Active VSX Gateway (VS 0) replies to the query. An active Virtual System on a Standby VSX Gateway will not reply to the query.

If you want to query the active Virtual System on a Standby VSX Gateway, use the real IP address of the VSX cluster member.

SNMP Configurations

To Configure:	Run:
SNMP Default	1. set snmp agent on 2. set snmp mode default
SNMP mode VS	1. set snmp agent on 2. set snmp mode vs
SNMP direct-vs-access	1. set snmp agent on 2. set snmp mode vs 3. set snmp vs-direct-access on

Example SNMP queries for Virtual Systems

This section shows example SNMP queries.

To run an SNMP V3 query using the VSX (VS 0) IP address

In clish

1. Enable the SNMP agent for context VS 0

Run:

```
set snmp agent on
```

2. Add an SNMP user with permissions for VSs 2,15.

Run:

```
add snmp usm user admin security-level authNoPriv auth-pass-phrase  
abcd1234
```

```
set snmp usm user admin vsid 2,15
```

3. Set SNMP VS mode.

Run:

```
set snmp mode vs
```

4. Send the remote queries, where:

- vsid is the SNMP context name required by SNMP V3.
- The IP address is the management IP address of the VSX gateway or cluster.

For example (in expert mode):

```
snmpwalk -n vsid2 -v 3 -l authNoPriv -u admin -A abcd1234 192.0.2.5 ifDesc
```

```
snmpwalk -n vsid15 -v 3 -l authNoPriv -u admin -A abcd1234 192.0.2.5 sysName
```

192.0.2.5 is the IP address of the management server.

To run an SNMP V1/V2c query using the VSX (VS 0) IP Address

In clish:

1. Enable the SNMP agent for context VS 0:

Run: set snmp agent on

2. Enable SNMP V1/V2C

Run: set snmp agent-version any

3. Set the SNMP community:


```
set snmp community public read-only
set snmp community private read-write
```
4. Set SNMP VS mode


```
set snmp mode vs
```
5. Send remote queries, where:
 - The community has the VSID or Virtual System name as a suffix.
 - The IP address is the Management IP address of the VSX gateway or cluster.

For example, to query a Virtual System with the name "MY_VS" or has VSID "2", run in expert mode:

```
snmpwalk -v 1 -c public_2 192.0.2.5 ifDescr
snmpwalk -v 1 -c private_MY_VS 192.0.2.5 ifDescr
```

Communities with suffixes are created automatically. Community name collisions might occur in special cases, for example if we use these communities:

- Read-only community = private
- Read-write community = private_1

The communities' private_1, and private_1_1 will be automatically created for VSID 1. Private_1 is not a unique community. The community is ambiguous and using it will result unexpected behavior.

To run an SNMP query using the virtual device IP address

1. Enable the SNMP agent for context VS0:


```
set snmp agent on
```
2. Add an SNMP user:


```
add snmp usm user admin security-level authNoPriv auth-pass-phrase
abcd1234
```
3. Specify USM user permissions for Virtual-Devices:


```
set snmp usm user admin vsid 0-10
```
4. Set the SNMP community:


```
set snmp community public read-only
set snmp community private read-write
```
5. Set SNMP VS mode:


```
set snmp mode vs
```
6. Enable SNMP queries over virtual device interfaces:


```
set snmp vs-direct-access on
```
7. Send remote queries, where the IP address is the Virtual IP address of the virtual device.

In expert mode, run:

```
snmpwalk -v 1 -c public 192.0.2.81 ifDescr
snmpwalk -v 2c -c public 192.0.2.81 ifDescr
snmpwalk -v 1 -c private 192.0.2.82 ifDescr
snmpwalk -v 2c -c private 192.0.2.82 ifDescr
snmpwalk -v 3 -l authNoPriv -u admin -A abcd1234 192.0.2.83 ifDescr
```

Note -

- The SNMP community used in the queries is the same community that you configured earlier.
- Only the active virtual device is queried.

Important - SNMP traps are available only for VS 0

The VSX SNMP Tree

To get information from a virtual device (system, switch, or router), you must load the Check Point mib.

- The mib is on the VSX gateway (context VS 0) at: `$CPDIR/lib/snmp`
- Load `chkpnt.mib` into your SNMP Browser or management application
- The VSX OID is: `.1.3.6.1.4.1.2620.1.16`

Example Commands

In clish

- To enable the SNMP agent for context VS 0, run:
`set snmp agent on`
- To add an SNMP user, run:
`add snmp usm user admin security-level authNoPriv auth-pass-phrase abcd1234`
- To set USM user permissions for VS 0, run:
`set snmp usm user admin vsid 0`
- To set the SNMP community, run:
`set snmp community public read-only`

In expert mode:

- To run an SNMP V2c query for VSX status table, run:
`snmpwalk -m $CPDIR/lib/snmp/chkpnt.mib -c public -v 2c 192.0.2.83 vsxStatusTable`
In expert mode:
- To run an SNMP V3 query for the vsx memory usage table, run:
`snmpwalk -m $CPDIR/lib/snmp/chkpnt.mib -v 3 -l authNoPriv -u admin -A abcd1234 192.0.2.83 vsxStatusMemoryUsageTable`

Note - When you use `snmpwalk` command with the OID name, you must use the `-m` option to specify the MIB.

The vsxCountersTable Refresh time

The `vsxCountersTable` refresh time is configured in:

`$FWDIR/conf/amon_vsx_refresh_interval`

The default value is 30 (seconds).

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 SmartConsole.
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 SmartConsole.
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 SmartConsole.
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 SmartConsole.
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:

General Troubleshooting Steps	187
Troubleshooting Specific Problems	189

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 192).

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 `cphaprob 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 *R80.10 ClusterXL Administration Guide* <http://downloads.checkpoint.com/dc/download.htm?ID=54804> 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`, `tracert`, `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. SmartConsole gives an error message:

Certificate cannot be pushed. Connection error with wait agent.

Possible Causes	How to Resolve
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.	On all relevant machines, re-check the cables, routes, IP addresses and any intermediate networking devices (routers, switches, hubs, and so on) between the management and the gateway(s).
Check that all the Check Point processes on the VSX Gateway(s) are up and running by running <code>cpwd_admin list</code> 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 <code>application ... start</code> command. (For more information refer to the Crossbeam documentation.)
Check that the CPD process is listening to the trust establishment port.	Run <code>netstat -an grep 18211</code> on the VSX Gateway(s), and make sure that output looks like this: <pre>tcp 0 0 0.0.0.0:18211 0.0.0.0:* LISTEN</pre>

SIC Trust Problems with New virtual devices

When creating a new Virtual System, Virtual Router or Virtual Switch, you cannot establish SIC trust.

Possible Causes	How to Resolve
Time or time zone mismatch between the management and the gateway. For proper SIC operation, the time, date and time zone must be synchronized between the management server and Gateways/ cluster members. 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.	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.

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 SmartConsole, 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 SmartConsole.**

Possible Causes	How to Resolve
Missing or invalid license on the management server. Execute <code>cplic check</code> on the management server to verify that you have the required licenses.	Obtain and install the appropriate licenses.
Missing or invalid VSX Gateway / cluster licenses. Run <code>fw vsx stat</code> on all Gateways, and make sure that the output says Number of Virtual Systems allowed by license: is greater than 0 .	Obtain a VSX and install a valid license for each VSX Gateway or cluster member.

Possible Causes	How to Resolve
Time or time zone mismatch between the management and the gateway. For proper SIC operation, the time, date and time zone must be synchronized between the management server and Gateways / cluster members. 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.	Change the time, date and time zone on the management and/or the gateways so that their UTC/GMT times match. Refer to your operating system documentation for the exact commands needed to accomplish this.

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.

Possible Causes	How to Resolve
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.	Install a policy on the Virtual System that enables the traffic. Check with the SmartView Tracker that the Virtual System is allowing the traffic.
There is the VLAN configuration problem on a switch, or physical cable problem.	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.
Incorrect routing on adjacent routers or hosts.	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.
Incorrect IP address or net mask defined on the Virtual System VLAN interface.	Check the IP address and the net mask assigned to the Virtual System internal VLAN interface.

Command Line Reference

In This Section:

<code>vsx_provisioning_tool</code>	192
VSX Gaia CLI Commands.....	202
CoreXL Commands (fw ctl affinity)	205
Firewall Commands	209
VSX Commands.....	213
Link Aggregation CLI Commands.....	216
<code>vsx_util</code> Command	218
The <code>cphaprob</code> Command	229

`vsx_provisioning_tool`

Description This command adds or removes virtual devices, interfaces, and routes.

Run `vsx_provisioning_tool` on a Multi-Domain Server or Security Management Server.

Syntax `vsx_provisioning_tool [-h] [-s <server>] {-u <user> | -c <certificate>} -p <password> [-a] {-o <commands> | -f <inputFile> [-l <line>]} -L`

Parameter	Description
-h	Show usage
-s	IPv4 or IPv6 address, or DNS name, of the management server (Required if <code>vsx_provisioning_tool</code> runs from a remote computer or Domain Management Server)
-u	Administrator user name
-c	Path to administrator certificate file
-p	Password of the user name or certificate file
-o	Execute these commands (" <code>vsx_provisioning_tool</code> Commands" on page 194) on the command line
-f	Path to a file with the commands (" <code>vsx_provisioning_tool</code> Commands" on page 194) to run
-l	Line number in file to start from (Applied with -f flag)
-a	Require connectivity to all VSX computers before committing transaction
-L	Local authentication mode

Examples

This line runs the commands from a text file:

```
vsx_provisioning_tool -s localhost -u admin -p admin -f
corpvsvx.txt
```

This command creates a new Virtual System (VS1) on the VSX1 cluster and adds a VLAN interface to VS1:

```
vsx_provisioning_tool -s localhost -u user -p pwd -o add vd name
VS1 vsx VSX1 , add interface name eth4.100 ip 1.1.1.1/24
```

Comments

- You must use the `-o` option or the `-f` option.
If you use a file: lines that begin with a hash sign (`#`) are treated as comments.
- If you use the `-a` flag, `vsx_provisioning_tool` makes sure it can connect to all VSX computers. But this does not guarantee that all the VSX computers can successfully apply the changes.
- In Multi-Domain Security Management environments, the IP address of the server (`-s` option) must be the Domain Management Server of the Virtual Device to change. If you run `vsx_provisioning_tool` locally on a Multi-Domain Server, you can connect to the current Domain, as chosen by the `mdsenv` command. Enter an IP address of localhost, or `127.0.0.1`.

Exit Code

Return Code	Description
0	Successfully applied all changes, on all cluster members.
1	Successfully applied all changes to the management database, but not to all VSX members.
2	Successfully applied all changes, but SIC communication failed to establish with at least one cluster member.
3	Connectivity check failed with at least one cluster member (if <code>-a</code> flag used). No changes to the database or to the VSX machines applied.
4	Failed to apply changes (due to internal error, syntax error, or other).

If commands are executed from a file with multiple transactions, the exit code refers to the last transaction processed.

Transactions

A transaction is a set of operations done on one Virtual Device. All operations are committed to the database together when the transaction ends. If the transaction fails, all its commands are discarded.

Name the Virtual Device with a parameter in the first command (all commands have a parameter to name the Virtual Device). You do not need to name it again in other commands of the same transaction. You cannot send operations to different Virtual Devices in one transaction. You cannot start a new transaction until you exit the one before.

When you send commands with the `-o` option, you can enter multiple commands (for example: add a Virtual System and then add interfaces and routes to it). Separate the commands with a comma (,). All the commands are one transaction. Explicit transaction commands are not allowed with this flag.

When you send commands with the `-f` option and a text file, you can use explicit transaction commands ("[vsx_provisioning_tool Commands](#)" on page 194). Commands from a file can be one or more transactions:

- If not inside a transaction, the current line is one transaction, which is automatically committed. You can write multiple commands in one line (as one transaction), separated with a comma (,).
- If currently inside a transaction, the lines are processed, but no action is done until the transaction ends.

vsx_provisioning_tool Commands

All `vsx_provisioning_tool` commands are pairs of key and value. The first two words in each command must appear in the correct order. Other pairs can be given in any order.

You can write in-line comments with a hash (#). Everything after the hash is part of the comment.

Transaction Explicit Commands

Operation	Command Syntax
Begin a new transaction	<code>transaction begin</code>
End a transaction	<code>transaction end</code>
Cancel a transaction	<code>transaction cancel</code>



Note - SIC with the Virtual System is established automatically. If it fails, operations continue and the transaction returns error code **2**.

Adding a Virtual Device

```
add vd name <vd name> vsx <vsx name> [type {vs|vsbm|vr|vsw}] [vs_mtu <MTU>]
[instances <IPv4 CoreXL instances>] [instances6 <IPv6 CoreXL instances>] [main_ip
<IPv4>] [main_ip6 <IPv6>] [calc_topo_auto {true|false}]
```

Parameter	Value	Notes
vd name	Name of the Virtual Device	Required if this is the first command in a transaction
vsx	VSX name to which the Virtual Device will be added	Required
type	Type of Virtual Device: • vs – Virtual System • vsbm – Virtual System in Bridge Mode • vr – Virtual Router • vsw – Virtual Switch	Default = vs
vs_mtu	Global MTU value for all interfaces	Applicable only for vsbm and vsw Default = 1500

Parameter	Value	Notes
<code>instances</code>	Number of CoreXL instances for IPv4	Applicable only for <code>vs</code> and <code>vsbm</code> Default = 1
<code>instances6</code>	Number of CoreXL instances for IPv6	Applicable only for <code>vs</code> and <code>vsbm</code> Default = 1
<code>main_ip</code>	Main IPv4 of the Virtual Device	Applicable only for <code>vs</code> and <code>vr</code>
<code>main_ip6</code>	Main IPv6 of the Virtual Device	Applicable only for <code>vs</code> and <code>vr</code>
<code>calc_topo_auto</code>	If true (default), automatically calculates topology based on routes	Applicable only for <code>vs</code> and <code>vr</code>

Example:

```
vsx_provisioning_tool -s localhost -u user -p pwd -o add vd name VSW1 vsx VSX1 type vsw
```

Comments:

- If `main_ip` and `main_ip6` are not used, the IPv4 and IPv6 addresses of the first interface added to the new device become the main IP addresses.
- For a Virtual Switch, if you do not add a VLAN or physical interface in the same transaction, `vs_mtu` is ignored.

Deleting a Virtual Device

```
remove vd name <vd name>
```

After you delete a Virtual Device, you cannot have more commands in the same transaction.

You cannot delete a Virtual Device if:

- It is referenced by a policy rule.
- It is referenced by other objects.
- It is enabled for global use in a Multi-Domain Security Management environment.

Modifying Settings of a Virtual Device

```
set vd name <vd name> [vs_mtu <MTU>] [instances <IPv4 CoreXL instances>] [instances6 <IPv6 CoreXL instances>] [main_ip <IPv4>] [main_ip6 <IPv6>] [calc_topo_auto {true|false}]
```

Parameter	Value	Notes
name	Name of the Virtual Device	Required if this is the first command in a transaction
vs_mtu	Global MTU value for all interfaces	Applicable only for vsbm and vsw
instances	Number of CoreXL instances for IPv4	Applicable only for vs and vsbm
instances6	Number of CoreXL instances for IPv6	Applicable only for vs and vsbm
main_ip	Main IPv4 of the Virtual Device	Applicable only for vs and vr
main_ip6	Main IPv6 of the Virtual Device	Applicable only for vs and vr
calc_topo_auto	To run automatic topology calculation based on routes	Applicable only for vs and vr

Example:

```
vsx_provisioning_tool -s localhost -u user -p pwd -o set vd name VS1 instances 8 main_ip 192.0.2.6 calc_topo_auto false
```

Comments:

- To remove existing values of main_ip and main_ip6, set the value to empty. For example:
set vd name VS1 main_ip6 empty

Adding an Interface

```
add interface vd <vd name> {name <interface>|leads_to <VR/VSW>} {ip|ip6 <address>{/<prefix length> | netmask <netmask> | netmask6 <netmask> | prefix <IP prefix> | prefix6 <IP prefix> } [{propagate | propagate6 <true|false>}] [topology {external|internal_undefined|internal_this_network|internal_specific specific_group <name>}]} [mtu <MTU>]
```

Parameter	Value	Notes
vd	Name of the Virtual Device	Required if this is the first command in a transaction.
name	Name of the physical or VLAN interface	name or leads_to must be given, but not both.
leads_to	Name of the Virtual Router or Virtual Switch that this interface is connected to	Applicable only for Virtual System.

Parameter	Value	Notes
<code>ip ip6</code>	IPv4 or IPv6 of the interface	Can be followed by a slash and a prefix length parameter. Applicable only for Virtual System or Virtual Router.
<code>netmask netmask6</code> <code>prefix prefix6</code>	If the address does not have a slash-prefix length, you must enter a netmask or a prefix	Use <code>netmask6</code> or <code>prefix6</code> for IPv6. Applicable only for Virtual System or Virtual Router. For interfaces on a Virtual System leading to a Virtual Router, the netmask or prefix must be the maximum possible for that address family: 255.255.255.255
<code>propagate propagate6</code>	If true, propagate the related IPv4 or IPv6 route to adjacent Virtual Devices	Applicable only for Virtual System and VLAN or physical interfaces. Default = <code>false</code>
<code>topology</code>	Topology of the interface	Applicable only for Virtual System, Virtual Router, or Virtual System in Bridge Mode. Default for Virtual System, Virtual Router = <code>internal_this_network</code> Default for Virtual System in Bridge Mode = <code>internal_undefined</code> <code>internal_this_network</code> cannot be set for Virtual System in Bridge Mode.
<code>specific_group</code>	If <code>topology</code> <code>internal_specific</code> , name of the group network object which defines the topology	Applicable only if automatic topology calculation is off.
<code>mtu</code>	MTU of the interface	Applicable only for Virtual System or Virtual Router. Default = 1500

Example:

Add VLAN interface eth4.100 with IP 1.1.1.1/24 to virtual device VS1:

```
vsx_provisioning_tool-s localhost -u user -p pwd -o add interface vd VS1 name eth4.100 ip 1.1.1.1/24
```

Removing an Interface

```
remove interface vd <vd name> {name <interface> | leads_to <VR | VSW>}
```

Parameter	Value	Notes
vd	Name of the Virtual Device	Required if this is the first command in a transaction.
name	Name of the interface to remove	name or leads_to must be given. You can give both, but if you do, they must be consistent and identify the same interface.
leads_to	Name of the Virtual Router or Virtual Switch that this interface is connected to	Applicable only for Virtual System.

Example:

```
vsx_provisioning_tool -s localhost -u user -p pwd -o remove interface vd VS1 name eth4.100
```

Comments:

- If there are routes that have a next-hop IP which would become inaccessible without this interface, the transaction will fail.
- IMPORTANT – If the interface being removed leads to a Virtual Router, all routes through that interface will be removed automatically.

Modifying Settings of an Interface

```
set interface vd <vd name> {name <interface> [new_name <interface>]|leads_to <VR | VSW> [new_leads_to <VR | VSW>]} [{propagate|propagate6} {true|false}] [topology {external|internal_undefined|internal_this_network|internal_specific specific_group <name>}] [mtu <MTU>]
```

Parameter	Value	Notes
vd	Name of the Virtual Device	Required if this is the first command in a transaction
name	Name of the physical or VLAN interface to change	name or leads_to must be given if you give both, they must identify the same interface
leads_to	Name of the Virtual Router or Virtual Switch that this interface is connected to	Applicable only for Virtual System
new_name	New name for the interface	You can change the name, but not the type

Parameter	Value	Notes
<code>new_leads_to</code>	New Virtual Router or Virtual Switch that this interface will connect to	of interface: - VLAN/physical interface can only be changed to a VLAN/physical interface - Interface leading to a VSW can only be changed to lead to a different VSW. - Interface leading to a VR can only be changed to lead to a different VR.
<code>propagate</code> <code>propagate6</code>	If true, propagate the related IPv4 or IPv6 route to adjacent Virtual Devices	Applicable only for Virtual System, and for VLAN or physical interfaces
<code>topology</code>	Topology of the interface	Applicable only for Virtual System, Virtual Router, or Virtual System in Bridge Mode <code>internal_this_network</code> cannot be set for Virtual System in Bridge Mode If <code>internal_specific</code> , enter <code>specific_group</code> with the name of the group network object which defines the topology (<code>specific_group</code> is applicable only if automatic topology calculation is off)
<code>mtu</code>	MTU of the interface	Applicable only for Virtual System or Virtual Router

Example:

```
vsx_provisioning_tool -s localhost -u user -p pwd -o set interface vd VS1 name
eth4.100 new_name eth5 propagate true topology internal_specific specific_group
NYGWs
```

Comments:

You cannot change or remove the IP or netmask of an existing interface with this command. You can remove the interface and add a new interface with a different IP, but not all the previous interface settings will be kept.

Adding a Route

This command is applicable only for Virtual System and Virtual Router.

```
add route vd <vd name> destination {IP [/prefix] | default | default6} [{netmask
<netmask> | prefix <prefix length>}] {next_hop <next hop ip> | leads_to <VR | VS>}
[propagate {true|false}]
```

Parameter	Value	Notes
<code>vd</code>	Name of the Virtual Device	Required if this is the first command in a transaction
<code>destination</code>	IP address of the route destination, can have slash-prefix length	IP version (IPv4 or IPv6) is automatically detected

Parameter	Value	Notes
netmask	Netmask of the destination	If an IP address (not default) is given without a prefix, netmask or prefix must be given
prefix	Prefix length of the destination	
next_hop	Next hop IP address for the route	next_hop IP address must be on a subnet of an existing interface
leads_to	Name of the Virtual Router or Virtual System which will be the next hop of the route	next_hop or leads_to must be given, but not both
propagate	If true, propagate this route to adjacent Virtual Devices.	Applicable only if next_hop is given Default = false

Example:

```
vsx_provisioning_tool -s localhost -u user -p pwd -o add route vd VS1 destination default leads_to VR3
```

Removing a Route

This command is applicable only for Virtual System and Virtual Router.

```
remove route vd <vd name> destination {IP[/prefix] | default | default6} [{netmask <netmask> | prefix <prefix length>}]
```

Parameter	Value	Notes
vd	Name of the Virtual Device	Required if this is the first command in a transaction
destination	IP address of the route destination, can have slash-prefix length	IP version (IPv4 or IPv6) is automatically detected
netmask	Netmask of the destination	If an IP address (not default) is given without a prefix, netmask or prefix must be given
prefix	Prefix length of the destination	

Example:

```
vsx_provisioning_tool -s localhost -u user -p pwd -o remove route vd VS1 destination default6
```

Showing Virtual Device Data

```
show vd name <vd name>
```

Parameter	Value	Notes
vd name	Name of the Virtual Device	Required

Comments:

- Only non-automatic routes are displayed. Routes which are automatically created with route propagation are not displayed.
- For Virtual Router and Virtual Switch: automatically created wrpj interfaces which connect to a Virtual System are not displayed.

Script Examples

Create a Virtual System connected to a Virtual Router, with a default route on the Virtual System to the Virtual Router. Add routes on the Virtual Router going to the Virtual System.

```
transaction begin
add vd name VR1 vsx VSX1 type vr
add interface name eth3.100 ip 10.0.0.1/24
transaction end

transaction begin
add vd name VR2 vsx VSX2 type vr
add interface name eth3.200 ip 20.0.0.1/24
transaction end

transaction begin
add vd name VS1 vsx VSX1
add interface leads_to VR1 ip 192.168.1.1/32
add interface name eth4.20 ip 192.168.20.1/24 propagate true
add route destination default leads_to VR1
add route destination 192.168.40.0/25 next_hop 192.168.20.254
transaction end
```

Create a Virtual System connected to a Virtual Switch, with manual topology.

```
transaction begin
add vd name VSW1 vsx VSX1 type vsw vs_mtu 1400
add interface name eth3.100
transaction end

transaction begin
add vd name VS1 vsx VSX1 calc_topo_auto false
add interface leads_to VSW1 ip 10.0.0.1/24 ip6 2001::1/64 topology external
add interface name eth4.20 ip 192.168.20.1/25 ip6 2020::1/64 topology
internal_this_network
add route destination default next_hop 10.0.0.254
add route destination default6 next_hop 2001::254
transaction end
```

Add CoreXL instances to the Virtual System made in the last example. Turn on automatic calculation of topology. Change the name of the internal interface, and reduce its MTU.

```
transaction begin
set vd name VS1 instances 4 instances6 2 calc_topo_auto true
set interface name eth4.20 new_name eth4.21 mtu 1400
transaction end
```

VSX Gaia CLI Commands

This section describes how to use Gaia CLI commands on VSX Gateways. For more about Gaia CLI commands, see the *R80.10 Gaia Administration Guide*

<http://supportcontent.checkpoint.com/solutions?id=sk111841>.

Changing the Context

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.

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>
```

Parameter	Description
<code>vsid</code>	ID of the virtual device.

Example

```
set virtual-system 3
```

Output

```
Context is set to vsid 3
```

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.

R80.10 VSX does not support the Gaia WebUI. You cannot configure role permissions for the WebUI.

For more about configuring roles, see the *R80.10 Gaia Administration Guide*

<http://downloads.checkpoint.com/dc/download.htm?ID=54824>.

Syntax

```
add rba role <name> virtual-system-access <vsid>
delete rba role <name> virtual-system-access <vsid|all>
```

Parameter	Description
name	Role name as a character string that contains letters, numbers or the underscore (_) character. The role name must start with a letter.
vsid	ID of the virtual device. Use commas to enter multiple virtual devices.
all	Deletes access permissions to all virtual devices.

Example

```
add rba role newRole domain-type System all-features
add rba role newRole virtual-system-access 0,2,3
```

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 204), 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.

```
GizaVSXGW1:0> add rba role newRole domain-type System all-features
GizaVSXGW1:0> add rba role newRole virtual-system-access 0,2,3
GizaVSXGW1:0> save config
```

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>
```

Parameter	Description
name	Role name as a character string that contains letters, numbers or the underscore (_) character. The role name must with a letter.
vsid	ID of the virtual device. Use commas to enter multiple virtual devices.
all	Deletes access permissions to all virtual devices.

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

```
Virtual Systems list
VS ID      VS NAME
0          VSX_GW
1          VSX_GW1_VSW
2          VS1
3          VS2
```

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 with the `-d` parameter, VSX automatically creates or updates the affinity configuration files.



Note - The `fw ctl affinity` command is different for a VSX Gateway and a Security Gateway:

VSX Gateway - Use the `-d` parameter to save the CoreXL affinity settings after you reboot it

Security Gateway - The CoreXL affinity settings are not saved after you reboot it

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>}
```

Parameter	Description
<instances>	Number range of firewall instances that you are setting affinity. Use a dash to set a range of instances.
<cpus>	Number range of CPU processing cores that you are setting affinity. Use a dash to set a range of cores.
<cores>	Number of cores that are used for CPU affinity. You cannot use this parameter to assign specific cores to the firewall instances.

Example

```
vsenv 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>
```

Parameter	Description
<cores>	Number of cores that are used for CPU affinity. You cannot use this parameter to assign specific cores to the firewall instances.

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>
```

Parameter	Description
process	Name of process that you are setting affinity.
vsids	Virtual System IDs that you are setting affinity for this process. Use a dash to set a range of Virtual Systems.
cpus	Number range of CPU processing cores that you are setting affinity. Use a dash to set a range of cores.

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 `-vsid` parameter to set affinity to the specified Virtual Systems. If you do not use the `-vsid` parameter, the affinity of the current Virtual System is set.

Syntax

```
fw ctl affinity -s -d [-vsid <vsids>] -cpu <cpus>
```

Parameter	Description
vsids	Virtual System IDs that you are setting affinity. Use a dash to set a range of Virtual Systems.
cpus	Number range of CPU processing cores that you are setting affinity. Use a dash to set a range of cores.

Example

```
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 `-vsid` parameter to show the affinity for a process to the specified Virtual Systems.

Syntax

```
fw ctl affinity -l -x [-vsid <vsids>] [-flags [e|h|k|n|t]]
```

Parameter	Description
vsids	Shows the affinity for processes for these Virtual System IDs. Use a dash to set a range of Virtual Systems.
e	Do not show processes that are affinity exceptions. Affinity exceptions are configured in the <code>\$FWDIR/conf/vsaffinity_exception.conf</code> file.
h	Show CPU affinity mask in hexadecimal format.
k	Do not show kernel threads.
n	Show the process name instead of <code>/proc/<pid>/cmdline</code>
t	Show information about the process threads.

Example

```
fw ctl affinity -l -x -vsid 1 -flags tn
```

Output

PID	VSID	CPU	SRC	V	KT	EXC	NAME
4756	0	all					pm
4773	0	all					confd
4774	0	all					searchd
5008	0	all					---searchd
4780	0	all					httpd2
4781	0	all					monitord
24700	0	0 1	P				---cpd
24704	0	0 1	P				---cpd
24705	0	0 1	P				---cpd
22800	0	all					mpdaemon
24523	0	all					fwk_forker
24525	0	all					fwk_wd
24573	0	1 3 4 6	P				fw
24667	0	1 3 4 6	P				---fw
24668	0	1 3 4 6	P				---fw
24670	0	1 3 4 6	P				---fw
24671	0	1 3 4 6	P				---fw
25412	0	1 3 4 6	P				---fw
24642	0	2 3 4 5 6 7	P				fwk0_dev
24643	0	2 3 4 5 6 7	P				---fwk0_0
30186	0	all					clishd

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
```

Firewall Commands

This section presents the usage of standard firewall (fw) commands as applicable to VSX Gateways and Virtual Systems.

fw getifs

Description

Shows a driver interface list for a specific Virtual System. By default, the VSX Gateway interface is displayed.

Run `vsenv <vsid>` to change context and show an interface list for a different Virtual System.

Syntax

```
fw getifs
```

Return Value

0 (zero) indicates that the command executed successfully. Any other response indicates an error.

Output

```
fw getifs
localhost vnd0 0.0.0.0 0.0.0.0
localhost eth0 4.4.6.101 255.255.0.0
localhost eth1 0.0.0.0 0.0.0.0
localhost sdp0 0.0.0.0 0.0.0.0
localhost int 0.0.0.0 0.0.0.0
localhost mgmt 10.18.83.171 255.255.255.0
localhost sync 7.7.7.171 255.255.255.0
localhost wrpj50001 0.0.0.0 0.0.0.0
localhost wrpj50003 0.0.0.0 0.0.0.0
```

fw monitor

Description

Captures network packets at multiple points within the VSX environment. You can only run one instance of this command at a time on VSX Gateway.

This section only presents the syntax relevant for VSX Gateways or clusters.

Syntax

```
fw monitor [-v vsid]
```

Parameters

Parameter	Description
[-v vsid]	Specify a gateway or Virtual System by its ID. The specific Virtual System on which packets should be captured. The default gives the VSX Gateway.

Return Value

0 (zero) indicates that the command executed successfully. Any other response indicates an error.

Example

```
fw monitor -v 2 -e 'accept ip_p=6' shows all TCP packets passing through Virtual System 2.
```

Output

```
[member1:0]# fw monitor
monitor: getting filter (from command line)
monitor: compiling
monitorfilter:
Compiled OK.
monitor: loading
monitor: monitoring (control-C to stop)
eth4:o[124]: 192.168.200.171 -> 192.168.200.1 (TCP) len=124 id=56430
TCP: 22 -> 1794 ...PA. seq=28d95f71 ack=57e454b1
eth4:o[124]: 192.168.200.171 -> 192.168.200.1 (TCP) len=124 id=56430
TCP: 22 -> 1794 ...PA. seq=28d95f71 ack=57e454b1
eth4:i[40]: 192.168.200.1 -> 192.168.200.171 (TCP) len=40 id=64876
TCP: 1794 -> 22 ....A. seq=57e454b1 ack=28d95fc5
eth4:I[40]: 192.168.200.1 -> 192.168.200.171 (TCP) len=40 id=64876
TCP: 1794 -> 22 ....A. seq=57e454b1 ack=28d95fc5
monitor: caught sig 2
monitor: unloading
```

fw tab

Description

Displays state tables for a specific Virtual System. State tables are used to store state information that Virtual Systems use to correctly inspect packets.

Run `vsenv <vsid>` to change context and show an interface list for a different Virtual System.

Syntax

```
fw tab [-t table name] [...]
```

Parameters

Parameter	Description
-t table name	Shows the state table for the specified Virtual System.
[...]	Arguments as defined for non-VSX machines.

Example

```
vsenv 1
fw tab -t connections
```

Output

```
localhost:
----- connections -----
dynamic, id 8158, attributes: keep, sync, expires 25,
refresh, limit 15000, hashsize 32768, kbuf 16 17 18
19 20 21 22 23 24 25 26 27 28 29 30, free function
90adc508 0

00000000, 0a125364, 00008a69, 0a12ae0a, 00004710,
00000006; 0001c001, 00804000, 08000000, 00000e10,
00000000, 3f7c2df6, 00000001, 66010202, 000007b6,
ffffffff, ffffffff, ffffffff, ffffffff, 00000000,
00000000, 00000000, 00000000, 00000000, 3bcd7000,
00000000, 00000000, 00000000; 3596/3600>
00000001, 0a12ae0a, 00004710, 0a125364, 00008a69,
00000006> 00000000, 0a125364, 00008a69, 0a12ae0a,
00004710, 00000006> 00000005)
00000000, 0a125364, 00008a6c, 0a12ab0a, 00004710,
00000006; 0001c001, 00804000, 08000000, 00000e10,
00000000, 3f7c2e4b, 00000001, 65060404, 000007b6,
0000000b, 0000000b, ffffffff, ffffffff, 00000000,
00000000, 00000000, 00000000, 00000000, 1fce4000,
00000000, 00000000, 00000000, 00000000, 00000000;
3581/3600>
00000001, 0a12ac0a, 00004710, 0a125364, 00008a6b,
00000006> 00000000, 0a125364, 00008a6b, 0a12ac0a,
00004710, 00000006> (00000005)
```

fw fetch

Description

Fetches the Inspection Code from the specified host and installs it to the kernel. Run `vsenv <vsid>` to change context and show an interface list for a different Virtual System.

Syntax

```
fw fetch [-n] [-f <filename>] [-c] [-i] master1 [master2] ...
```

Parameter	Description
-n	Fetch the Security Policy from the Security Management Server to the local <code>state</code> directory, and install the Policy only if the fetched Policy is different from the Policy already installed.
-f <filename>	Fetch the Security Policy from the Security Management Server listed in <filename>. If filename is not specified, the list in <code>conf/masters</code> is used.
-c	Cluster mode, get policy from one of the cluster members, from the Check Point High Availability (CPHA) kernel list.
-i	Ignore SIC information (for example, SIC name) in the database and use the information in <code>conf/masters</code> . This option is used when a Security Policy is fetched for the first time by a DAIP gateway from a Security Management Server with a changed SIC name.
master1	Runs the command on the designated master. The name of the Security Management Server from which to fetch the Policy. You may specify a list of one or more Security Management Servers, such as <code>master1 master2</code> which will be searched in the order listed. If <code>targets</code> is not specified, or inaccessible, the policy is fetched from <code>localhost</code> .

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

Parameter	Description
-c	Cluster mode
-n	Do not run <code>local.vsa11</code> if VSX configuration, as fetched from management server, is up-to-date.
-s	Concurrent fetches for multi-processor environment.
-q	Quiet mode - Only summary lines appear.
-v	Verbose mode - Detailed information appears.
-f conf_file	Fetches NCS commands configuration file instead of the default <code>local.vsa11</code> .
local	Reads <code>local.vsa11</code> configuration file from <code>\$FWDIR/state/local/vsx</code> and executes the NCS
management	Fetches <code>local.vsa11</code> 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.

Output

```
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.vskip).

Purge operation succeeded.
Fetching Virtual Systems configuration file (local.vsal).

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

Parameter	Description
-q	Quiet output. Only summary information appears.
-v	Verbose output. Detailed information appears.
<vs name> <vs ID>	Enter the Virtual System name or ID.

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

Parameter	Description
-v	Displays detailed (verbose) information.
-l	Displays a detailed list of all virtual devices
<vsid>	Displays statistics for the specified Virtual System

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-FW08.0PA
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

```
vsenv 1
vsx 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 SmartConsole, 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
```

Bond name: bond0
 Bond mode: Load Sharing
 Bond status: Up
 Balancing mode: 802.3ad Layer3+4 Load Balancing
 Configured slave interfaces: 4
 In use slave interfaces: 4
 Required slave interfaces: 2

Slave Name	Status	Link
eth2	Active	Yes
eth3	Active	Yes
eth4	Active	Yes
eth5	Active	Yes

Report Results

- Required slave interfaces ("[Setting Critical Required Interfaces](#)" on page 154).
- 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

```
[Expert@mem97:0]# cphaprob -a if

vsid 0:
-----
Required interfaces: 7
Required secured interfaces: 1

eth0      UP                non sync(non secured), multicast
eth1      UP                sync(secured), broadcast

Virtual cluster interfaces: 1

eth0      192.0.2.98
          2001:db8:0:a::98

vsid 1:
-----
Required interfaces: 3
Required secured interfaces: 1

eth1      UP                sync(secured), broadcast
eth2      UP                non sync(non secured), multicast
eth3      UP                non sync(non secured), multicast

Virtual cluster interfaces: 2

eth2      10.2.2.2
          2001:db8:0:b::1
eth3      172.16.3.3
          2001:db8:0:c::3121

vsid 2:
-----
Required interfaces: 2
Required secured interfaces: 1

eth1      UP                sync(secured), broadcast
eth5      UP                non sync(non secured), multicast

Virtual cluster interfaces: 1

eth5      192.168.5.5
          2001:db8:0:d::1
```

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).

Syntax vsx_util <sub-command> [parameters]

Parameters

Parameter	Description
-s <management IP>	Perform action using the specified management IP
-u <user name>	Perform the action using the specified administrator
-c <cluster or gateway name>	Perform the action on the specified cluster or VSX Gateway
-m <member name>	Perform the action on the specified member
-h	Display help text

Comments **Note** - You must close SmartConsole 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.

Syntax `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)

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 125) before using this command.

add_member_reconf

Description	Restores VSX configuration after adding a cluster member
Syntax	<code>vsx_util add_member_reconf</code>
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 125) 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	<code>vsx_util change_interfaces</code>
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 56) from an Open Server to a Check Point appliance by using the Management Only mode. - Refer to the notes ("Notes" on page 221) for additional information.
	Important - You must close SmartConsole 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:

1. Close SmartConsole for the Security Management Server and/or Multi-Domain Security Management Domain Management Servers.
2. On the management server, enter the Expert Mode.
3. Run `vsx_util change_interfaces`
4. Enter the IP address for the Security Management Server or Multi-Domain Security Management main Domain Management Server.
5. Enter the administrator name and password as requested.
6. Enter or select the VSX cluster or VSX Gateway object name.

7. 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 224) to push the updated configuration to VSX Gateways or cluster members.
8. Select the interface to be replaced.
9. 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).
10. To replace additional interfaces, enter **y** when prompted and repeat steps 6 through 8.
11. To complete the process, enter **n**.
12. If you selected the **Apply changes to management only** option, you can remove the replaced interfaces from the database. Enter "**y**" for this option:
 Would you like to remove the old interfaces from the database? (y|n) [n]:
13. 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 224) 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 within the same subnet. See sk92425 http://supportcontent.checkpoint.com/solutions?id=sk92425 .
Syntax	<code>vsx_util change_mgmt_ip</code>
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_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 56) from an Open Server to a Check Point appliance by using the **Management Only** mode.
 - Refer to the notes ("**Notes**" on page 221) for additional information.



Important - You must close SmartConsole 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:

1. Close SmartConsole for the Security Management Server and/or Multi-Domain Security Management Domain Management Servers.
2. On the management server, enter the Expert Mode.
3. Run `vsx_util change_interfaces`
4. Enter the IP address for the Security Management Server or Multi-Domain Security Management main Domain Management Server.
5. Enter the administrator name and password as requested.
6. Enter or select the VSX cluster or VSX Gateway object name.
7. 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 224) to push the updated configuration to VSX Gateways or cluster members.
8. Select the interface to be replaced.

9. 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).
10. To replace additional interfaces, enter **y** when prompted and repeat steps 6 through 8.
11. To complete the process, enter **n**.
12. If you selected the **Apply changes to management only** option, you can remove the replaced interfaces from the database. Enter **"y"** for this option:
Would you like to remove the old interfaces from the database? (y|n) [n]:
13. 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 224) 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 IP address to a different subnet. See sk92425 <http://supportcontent.checkpoint.com/solutions?id=sk92425>.

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.

change_private_net

Description	Changes the cluster internal communication network IP address.
Syntax	<code>vsx_util change_private_net</code>
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. For an IPv4 cluster, the default cluster private network uses 255.255.252.0 for the netmask. You can change this value. For an IPv6 cluster, the new cluster private network must use /80 for the netmask. Run the command and follow the instructions on the screen.

convert_cluster

Description	Converts the cluster type from High Availability to VSLS or from VSLS to High Availability
Syntax	<code>vsx_util convert_cluster</code>
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	<code>vsx_util reconfigure</code>
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

Option	Description
1) All Interfaces	Show all interfaces (physical and Warp)
2) All Physical Interfaces	Show Physical interfaces only
3) All Warp Interfaces	Show Warp interfaces only
4) A Specific Interface	Enter the interface name when prompted to a specific interface.



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

```
Expert@mgmt95:0]# vsx_util show_interfaces
Enter Security Management Server/main Domain Management Server IP address
(Hit 'ENTER' for 'localhost'):
Enter Administrator Name: jon
Enter Administrator Password:

Select VSX Gateway/cluster object name:
  1) vsx-cluster
  2) vsx-cluster_2
  3) vsx-gw
  4) vsx-gw_2
Select: 1

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: 1
```

Type & Interface	Virtual Device Name	VSID	IP / Mask length
M eth0	vsx-cluster	0	v4 172.16.16.98/24 v6 2001:0DB8::98/64
S eth1	vsx-cluster	0	v4 10.0.0.0/24
U eth2	vs1	1	v4 192.0.2.2/24 v6 2001:0DB8:c::1/64
U eth3	vs1	1	v4 192.168.3.3/24 v6 2001:0DB8:b::1/64
A eth4			
U eth5	vs2	2	v4 10.10.10.10/24 v6 2001:0DB8:a::1/64
A eth6			

```
#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 vsx_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 `vsx_util view_vs_conf`

Output

```
[Expert@mgmt95:0]# vsx_util view_vs_conf
Enter Security Management Server/main Domain Management Server IP address (
Hit 'ENTER' for 'localhost'):
Enter Administrator Name: jon
Enter Administrator Password:

Select VSX gateway/cluster object name:
1)    vsx-cluster
2)    vsx_cluster_2
3)    vsx-gw
4)    vsx-gw_2
Select: 1

Select Virtual Device object name:
1)    vs1
2)    vs2
3)    vs3
4)    vsx-cluster
Select: 1

Type: Virtual System

Interfaces configuration table:

+-----+-----+-----+-----+
|Interfaces|          |Mgmt|VSX GW(s)|
+-----+-----+-----+-----+
|Name      |IP / Mask length|mem96|mem97|
+-----+-----+-----+-----+
|eth2      |v4 10.0.0.0/24 v6 2001:db8::abc::1/64|V|V|V|
|eth3      |v4 10.10.10.10/24 v6 2001:db8::3121/64|V|V|V|
+-----+-----+-----+-----+

Interfaces Table Legend:

V - Interface exists on the gateway and matches management information
(if defined on the management).
- - Interface does not exist on the gateway.
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:

Ipv4 Routes			Mgmt	VSX GW(s)	
Destination / Mask Length	Gateway	Interface		mem96	mem97
2.2.2.0/24		eth2	V	V	V
3.3.3.0/24		eth3	V	V	V
Ipv6 Routes			Mgmt	VSX GW(s)	
Destination / Mask Length	Gateway	Interface		mem96	mem97
2001:db8::abc::/64		eth2	V	!NH	!NH
2001:db8:0a::/64		eth3	V	!NH	!NH
2001:db8::1ffe:0:0:0/112		eth2	-	V	V
2001:db8::fd9a:0:1:0/112		eth3	-	V	V

Routing Table Legend:

```
V - Route exists on the gateway and matches management information
(if defined on the management).
```

- Route does not exist on the gateway.

```
N/A - Fetching Virtual Device configuration from the gateway failed.
```

```
!NH - Route exists on the gateway, but there is a Next Hop mismatch.
```

Note: Routes can be created automatically on the gateways by the Operating System.

Therefore, routes that appear on all gateways, but are not defined on the management, do not necessarily indicate a problem.

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 132) 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 VSL 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_util vsls` command:

1. Run `vsx_util vsls` from the Expert mode on the management server
2. Select the desired choice from the **VSLS** menu

The cphaprob Command

You use the `cphaprob` command to verify cluster functionality and to debug cluster related problems. This section provides a brief overview of the `cphaprob` command and its command options. For complete documentation and use cases, refer to the *R80.10 ClusterXL Administration Guide* <http://downloads.checkpoint.com/dc/download.htm?ID=54804>.

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:

Cluster interfaces on the cluster members.

Synchronization — full synchronization completed successfully.

Filter — the Security Policy, and whether it is loaded.

fw — the VPN-1 daemon.

You can include these commands in scripts for automatic execution.

To produce a usage printout for `cphaprob` that shows all the available commands, type `cphaprob` at the command line and press **Enter**. The following output appears:

```
cphaprob state
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] syncstat . Sync transport layer statistics
cphaprob fcustat ..... Full connectivity upgrade statistics
cphaprob tablestat ..... Cluster tables
```

Command Options

This table describes the available command options:

Parameter	Description
<code>[-vs vsid] stat</code>	View the status all cluster members or for a specific Virtual System. -vs is relevant only for VSLS.
<code>[-a] [-vs vsid] if</code>	View the state of the cluster member interfaces and the virtual cluster interfaces. -vs is relevant only for VSLS.
<code>[-i[a]] [-e] list</code>	View the list of critical devices on a cluster member, and of all the other machines in the cluster.
<code>-d <device> -t <timeout (sec)> -s <ok init problem> [-p] register</code>	Register <device> as a critical process, and add it to the list of devices that must be running for the cluster member to be considered active.
<code>-f <file> register</code>	Register all the user defined critical devices listed in <file>.
<code>-d <device> [-p] unregister</code>	Unregister a user defined <device> as a critical process. This means that this device is no longer considered critical.
<code>-a unregister</code>	Unregister all user defined devices
<code>-d <device> -s <ok init problem> > report</code>	Report the status of a user defined critical device to ClusterXL.
<code>[-reset] ldstat -vs</code>	View sync serialization statistics. -vs is relevant only for VSLS.
<code>[-reset] syncstat -vs</code>	View sync transport layer statistics. -vs is relevant only for VSLS.
<code>tablestat</code>	Displays interfaces and IP addresses for each cluster member