Important Information

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

Latest Documentation
The latest version of this document is at: http://supportcontent.checkpoint.com/documentation_download?ID=12265
For additional technical information, visit the Check Point Support Center (http://supportcenter.checkpoint.com).

Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>13 December 2011</td>
<td>Minor textual corrections</td>
</tr>
<tr>
<td>13 December 2011</td>
<td>Improvements made since first release</td>
</tr>
<tr>
<td>14 July 2011</td>
<td>First release of this document</td>
</tr>
</tbody>
</table>

Feedback
Check Point is engaged in a continuous effort to improve its documentation.
Please help us by sending your comments (mailto:cp_techpub_feedback@checkpoint.com?subject=Feedback on ClusterXL R75.20 Administration Guide).
## Contents

<table>
<thead>
<tr>
<th>Important Information</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction to ClusterXL</td>
<td>8</td>
</tr>
<tr>
<td>The Need for Gateway Clusters</td>
<td>8</td>
</tr>
<tr>
<td>ClusterXL Gateway Cluster Solution</td>
<td>8</td>
</tr>
<tr>
<td>How ClusterXL Works</td>
<td>9</td>
</tr>
<tr>
<td>The Cluster Control Protocol</td>
<td>9</td>
</tr>
<tr>
<td>Installation and Platform Support</td>
<td>9</td>
</tr>
<tr>
<td>ClusterXL Licenses</td>
<td>9</td>
</tr>
<tr>
<td>Clock Synchronization in ClusterXL</td>
<td>10</td>
</tr>
<tr>
<td>Clustering Definitions and Terms</td>
<td>10</td>
</tr>
<tr>
<td>Synchronizing Connection Information Across the Cluster</td>
<td>11</td>
</tr>
<tr>
<td>The Check Point State Synchronization Solution</td>
<td>11</td>
</tr>
<tr>
<td>The Synchronization Network</td>
<td>11</td>
</tr>
<tr>
<td>How State Synchronization Works</td>
<td>12</td>
</tr>
<tr>
<td>Non-Synchronized Services</td>
<td>12</td>
</tr>
<tr>
<td>Configuring Services not to Synchronize</td>
<td>12</td>
</tr>
<tr>
<td>Duration Limited Synchronization</td>
<td>13</td>
</tr>
<tr>
<td>Non-Sticky Connections</td>
<td>13</td>
</tr>
<tr>
<td>Non-Sticky Connection Example: TCP 3-Way Handshake</td>
<td>14</td>
</tr>
<tr>
<td>Synchronizing Non-Sticky Connections</td>
<td>14</td>
</tr>
<tr>
<td>Synchronizing Clusters on a Wide Area Network</td>
<td>15</td>
</tr>
<tr>
<td>Synchronized Cluster Restrictions</td>
<td>15</td>
</tr>
<tr>
<td>Configuring State Synchronization</td>
<td>15</td>
</tr>
<tr>
<td>Configuring a Service Not to Synchronize</td>
<td>15</td>
</tr>
<tr>
<td>Creating Synchronized and Non-Synchronized Versions</td>
<td>16</td>
</tr>
<tr>
<td>Configuring Duration Limited Synchronization</td>
<td>16</td>
</tr>
<tr>
<td>Sticky Connections</td>
<td>17</td>
</tr>
<tr>
<td>Introduction to Sticky Connections</td>
<td>17</td>
</tr>
<tr>
<td>The Sticky Decision Function</td>
<td>17</td>
</tr>
<tr>
<td>VPN Tunnels with 3rd Party Peers and Load Sharing</td>
<td>18</td>
</tr>
<tr>
<td>Third-Party Gateways in Hub and Spoke Deployments</td>
<td>18</td>
</tr>
<tr>
<td>Configuring the Sticky Decision Function</td>
<td>19</td>
</tr>
<tr>
<td>Establishing a Third-Party Gateway in a Hub and Spoke Deployment</td>
<td>20</td>
</tr>
<tr>
<td>High Availability and Load Sharing in ClusterXL</td>
<td>22</td>
</tr>
<tr>
<td>Introduction to High Availability and Load Sharing</td>
<td>22</td>
</tr>
<tr>
<td>Load Sharing</td>
<td>22</td>
</tr>
<tr>
<td>High Availability</td>
<td>23</td>
</tr>
<tr>
<td>Example ClusterXL Topology</td>
<td>23</td>
</tr>
<tr>
<td>Defining the Cluster Member IP Addresses</td>
<td>24</td>
</tr>
<tr>
<td>Defining the Cluster Virtual IP Addresses</td>
<td>24</td>
</tr>
<tr>
<td>The Synchronization Network</td>
<td>24</td>
</tr>
<tr>
<td>Configuring Cluster Addresses on Different Subnets</td>
<td>25</td>
</tr>
<tr>
<td>ClusterXL Modes</td>
<td>25</td>
</tr>
<tr>
<td>Load Sharing Multicast Mode</td>
<td>25</td>
</tr>
<tr>
<td>Load Sharing Unicast Mode</td>
<td>26</td>
</tr>
<tr>
<td>High Availability Mode</td>
<td>27</td>
</tr>
<tr>
<td>Mode Comparison Table</td>
<td>28</td>
</tr>
<tr>
<td>Failover</td>
<td>29</td>
</tr>
<tr>
<td>When Does a Failover Occur?</td>
<td>29</td>
</tr>
<tr>
<td>What Happens When a Gateway Recovers?</td>
<td>29</td>
</tr>
<tr>
<td>How a Recovered Cluster Member Obtains the Security Policy</td>
<td>30</td>
</tr>
<tr>
<td>Implementation Planning Considerations</td>
<td>30</td>
</tr>
</tbody>
</table>
ClusterXL Advanced Configuration

Working with VPNs and Clusters
Configuring VPN and Clusters
Defining VPN Peer Clusters with Separate Security Management Servers
Working with NAT and Clusters
Cluster Fold and Cluster Hide
Configuring NAT on the Gateway Cluster
Configuring NAT on a Cluster Member

Working with VLANS and Clusters
VLAN Support in ClusterXL
Connecting Several Clusters on the Same VLAN
Monitoring the Interface Link State
Enabling Interface Link State Monitoring

Link Aggregation and Clusters
Overview
Link Aggregation - High Availability Mode
Link Aggregation - Load Sharing Mode
Defining VLANs on an Interface Bond
Performance Guidelines for Link Aggregation
ClusterXL Commands for Interface Bonds
Troubleshooting Bonded Interfaces

Advanced Cluster Configuration
How to Configure Gateway Configuration Parameters
How to Configure Gateway to Survive a Boot
Setting Module Variables in IPSO 6.1 and Later
Controlling the Clustering and Synchronization Timers
Blocking New Connections Under Load
Working with SmartView Tracker Active Mode
Reducing the Number of Pending Packets
Configuring Full Synchronization Advanced Options

Defining Disconnected Interfaces
Defining a Disconnected Interface on Unix
Defining a Disconnected Interface on Windows

Configuring Policy Update Timeout
Enhanced 3-Way TCP Handshake Enforcement

Configuring Cluster Addresses on Different Subnets
Introduction to Cluster Addresses on Different Subnets
Configuration of Cluster Addresses on Different Subnets
Example of Cluster Addresses on Different Subnets
Limitations of Cluster Addresses on Different Subnets

Moving from a Single Gateway to a ClusterXL Cluster
On the Single Gateway Machine
On Machine 'B'
In SmartDashboard, for Machine 'B'
Chapter 1

Introduction to ClusterXL

In This Chapter

The Need for Gateway Clusters 8
ClusterXL Gateway Cluster Solution 8
How ClusterXL Works 9
Installation and Platform Support 9
ClusterXL Licenses 9
Clock Synchronization in ClusterXL 10
Clustering Definitions and Terms 10

The Need for Gateway Clusters

Gateways and VPN connections are business critical devices. The failure of a Security Gateway or VPN connection can result in the loss of active connections and access to critical data. The gateway between the organization and the world must remain open under all circumstances.

ClusterXL Gateway Cluster Solution

A ClusterXL cluster is a group of identical Check Point Security Gateways connected in such a way that if one fails, another immediately takes its place.

ClusterXL is a software-based Load Sharing and High Availability solution that distributes network traffic between clusters of redundant Security Gateways and provides transparent failover between machines in a cluster.

- A High availability cluster ensures gateway and VPN connection redundancy by providing transparent failover to a backup gateway in the event of failure.
- A Load Sharing cluster provides reliability and also increases performance, as all cluster members are active.

Figure 1-1 Gateway Cluster
How ClusterXL Works

ClusterXL uses unique physical IP and MAC addresses for the cluster members and virtual IP addresses to represent the cluster itself. Virtual IP addresses do not belong to an actual machine interface (except in High Availability Legacy mode, explained later).

ClusterXL provides an infrastructure that ensures that data is not lost due to a failure, by ensuring that each cluster member is aware of connections passing through the other members. Passing information about connections and other Security Gateway states between the cluster members is known as State Synchronization.

Security Gateway Clusters can also be built using OPSEC certified High Availability and Load Sharing products. OPSEC certified clustering products use the same State Synchronization infrastructure as ClusterXL.

The Cluster Control Protocol

The Cluster Control Protocol (CCP) is the glue that links together the machines in the Check Point Gateway Cluster. CCP traffic is distinct from ordinary network traffic and can be viewed using any network sniffer.

CCP runs on UDP port 8116, and has the following roles:

- It allows cluster members to report their own states and learn about the states of other members by sending keep-alive packets (this only applies to ClusterXL clusters).
- State Synchronization.

The Check Point CCP is used by all ClusterXL modes as well as by OPSEC clusters. However, the tasks performed by this protocol and the manner in which they are implemented may differ between clustering types.

Note - There is no need to add a rule to the Security Policy Rule Base that accepts CCP

Installation and Platform Support

ClusterXL must be installed in a distributed configuration in which the Security Management server and the cluster members are on different machines. ClusterXL is part of the standard Security Gateway installation.

For more detailed installation instructions, see the R75.20 Installation and Upgrade Guide (http://supportcontent.checkpoint.com/documentation_download?ID=12269).

See the R75.20 Release Notes (http://supportcontent.checkpoint.com/documentation_download?ID=12414) for the ClusterXL supported platforms.

ClusterXL Licenses

To use ClusterXL for High Availability, each gateway in the configuration must have a regular gateway license and the management machine must have a license for each cluster defined.

To use ClusterXL for Load Sharing, each gateway in the configuration must have a regular gateway license and the management machine must have a license for each cluster defined and one additional cluster-1 primitive license.

It does not matter how many gateways are included in the cluster. If the proper licenses are not installed, the install policy operation will fail.

For more information about licenses, visit the Check Point User Center (http://usercenter.checkpoint.com).
Clock Synchronization in ClusterXL

When using ClusterXL, make sure to synchronize the clocks of all of the cluster members. You can synchronize the clocks manually or using a protocol such as NTP. Features such as VPN only function properly when the clocks of all of the cluster members are synchronized.

Clustering Definitions and Terms

Different vendors give different meanings to terms that relate to Gateway Clusters, High Availability, and Load Sharing. Check Point uses the following definitions and terms when discussing clustering:

**Active Up** - When the High Availability machine that was Active and suffered a failure becomes available again, it returns to the cluster, not as the Active machine but as one of the standby machines in the cluster.

**Cluster** - A group of machines that work together to provide Load Sharing and/or High Availability.

**Critical Device** - A device that the Administrator has defined to be critical to the operation of the cluster member. A critical device is also known as a Problem Notification (pnote). Critical devices are constantly monitored. If a critical device stops functioning, this is defined as a failure. A device can be hardware or a process. The `fwd` and `cphad` processes are predefined by default as critical devices. The Security Policy is also predefined as a critical device. The Administrator can add to the list of critical devices using the `cphaprob` command.

**Failure** - A hardware or software problem that causes a machine to be unable to filter packets. A failure of an Active machine leads to a Failover.

**Failover** - A machine taking over packet filtering in place of another machine in the cluster that suffered a failure.

**High Availability** - The ability to maintain a connection when there is a failure by having another machine in the cluster take over the connection, without any loss of connectivity. Only the Active machine filters packets. One of the machines in the cluster is configured as the Active machine. If a failure occurs on the Active machine, one of the other machines in the cluster assumes its responsibilities.

**Hot Standby** - Also known as Active/Standby. It has the same meaning as High Availability.

**Load Sharing** - In a Load Sharing Gateway Cluster, all machines in the cluster filter packets. Load Sharing provides High Availability, gives transparent Failover to any of the other machines in the cluster when a failure occurs, and provides enhanced reliability and performance. Load Sharing is also known as Active/Active.

**Multicast Load Sharing** - In ClusterXL's Load Sharing Multicast mode, every member of the cluster receives all of the packets sent to the cluster IP address. A router or Layer 3 switch forwards packets to all of the cluster members using multicast. A ClusterXL decision algorithm on all cluster members decides which cluster member should perform enforcement processing on the packet.

**Unicast Load Sharing** - In ClusterXL's Load Sharing Unicast mode, one machine (the Pivot) receives all traffic from a router with a unicast configuration and redistributes the packets to the other machines in the cluster. The Pivot machine is chosen automatically by ClusterXL.
Chapter 2

Synchronizing Connection Information Across the Cluster

In This Chapter

The Check Point State Synchronization Solution 11
Configuring State Synchronization 15

The Check Point State Synchronization Solution

A failure of a firewall results in an immediate loss of active connections in and out of the organization. Many of these connections, such as financial transactions, may be mission critical, and losing them will result in the loss of critical data. ClusterXL supplies an infrastructure that ensures that no data is lost in case of a failure, by making sure each gateway cluster member is aware of the connections going through the other members. Passing information about connections and other Security Gateway states between the cluster members is called State Synchronization.

Every IP based service (including TCP and UDP) recognized by the Security Gateway is synchronized. State Synchronization is used both by ClusterXL and by third-party OPSEC-certified clustering products.

Machines in a ClusterXL Load Sharing configuration must be synchronized. Machines in a ClusterXL High Availability configuration do not have to be synchronized, though if they are not, connections will be lost upon failover.

The Synchronization Network

The Synchronization Network is used to transfer synchronization information about connections and other Security Gateway states between cluster members.

Since the synchronization network carries the most sensitive Security Policy information in the organization, it is critical that you protect it against both malicious and unintentional threats. We recommend that you secure the synchronization interfaces using one of the following methods:

- Using a dedicated synchronization network
- Connecting the physical network interfaces of the cluster members directly using a cross-cable. In a cluster with three or more members, use a dedicated hub or switch

**Note** - You can synchronize members across a WAN by following the steps in Synchronizing Clusters on a WAN (see “Synchronizing Clusters on a Wide Area Network” on page 15).

Following these recommendations guarantees the safety of the synchronization network because no other networks carry synchronization information.

It is possible to define more than one synchronization network for backup purposes. It is recommended that the backup be a dedicated network.

In Cluster XL, the synchronization network is supported on the lowest VLAN tag of a VLAN interface. For example, if three VLANs with tags 10, 20 and 30 are configured on interface eth1, interface eth1.10 may be used for synchronization.
How State Synchronization Works

Synchronization works in two modes:

- **Full sync** transfers all Security Gateway kernel table information from one cluster member to another. It is handled by the `fwd` daemon using an encrypted TCP connection.
- **Delta sync** transfers changes in the kernel tables between cluster members. Delta sync is handled by the Security Gateway kernel using UDP multicast or broadcast on port 8116.

Full sync is used for initial transfers of state information, for many thousands of connections. If a cluster member is brought up after being down, it will perform full sync. After all members are synchronized, only updates are transferred via delta sync. Delta sync is quicker than full sync.

State Synchronization traffic typically makes up around 90% of all Cluster Control Protocol (CCP) traffic. State Synchronization packets are distinguished from the rest of CCP traffic via an opcode in the UDP data header.

**Note** - The source MAC address for CCP packets can be changed (see "Synchronizing Clusters on a Wide Area Network" on page 15).

Non-Synchronized Services

In a gateway cluster, all connections on all cluster members are normally synchronized across the cluster. Not all services that cross a gateway cluster must be synchronized.

- You can decide not to synchronize TCP, UDP and other service types. By default, all these services are synchronized.
- The VRRP and IP Clustering control protocols, and the IGMP protocol, are not synchronized by default (but you can choose to turn on synchronization for these protocols). Protocols that run solely between cluster members need not be synchronized. Although you can synchronize them, no benefit will be gained. This synchronization information will not help a failover. These protocols are not synchronized by default: IGMP, VRRP, IP clustering and some other OPSEC cluster control protocols.
- Broadcasts and multicasts are not synchronized, and cannot be synchronized.

You can have a synchronized service and a non-synchronized definition of a service, and use them selectively in the Rule Base.

Configuring Services not to Synchronize

Synchronization incurs a performance cost. You may choose not to synchronize a service if these conditions are true:

- A significant amount of traffic crosses the cluster through a particular service. Not synchronizing the service reduces the amount of synchronization traffic and so enhances cluster performance.
- The service usually opens short connections, whose loss may not be noticed. DNS (over UDP) and HTTP are typically responsible for most connections and frequently have short life and inherent recoverability in the application level. Services which typically open long connections, such as FTP, should always be synchronized.
- Configurations that ensure bi-directional stickiness for all connections do not require synchronization to operate (only to maintain High Availability). Such configurations include:
  - Any cluster in High Availability mode (for example, ClusterXL New HA or IPSO VRRP).
  - ClusterXL in a Load Sharing mode with clear connections (no VPN or static NAT).
  - OPSEC clusters that guarantee full stickiness (refer to the OPSEC cluster’s documentation).
  - VPN and Static NAT connections passing through a ClusterXL cluster in a Load Sharing mode (either multicast or unicast) may not maintain bi-directional stickiness. State Synchronization must be turned on for such environments.
Duration Limited Synchronization

Some TCP services (HTTP for example) are characterized by connections with a very short duration. There is no point in synchronizing these connections because every synchronized connection consumes gateway resources, and the connection is likely to have finished by the time a failover occurs.

For all TCP services whose Protocol Type (that is defined in the GUI) is HTTP or None, you can use this option to delay telling the Security Gateway about a connection, so that the connection will only be synchronized if it still exists x seconds after the connection is initiated. This feature requires a SecureXL device that supports "Delayed Notifications" and the current cluster configuration (such as Performance Pack with ClusterXL LS Multicast).

This capability is only available if a SecureXL-enabled device is installed on the Security Gateway through which the connection passes.

The setting is ignored if connection templates are not offloaded from the ClusterXL-enabled device. See the SecureXL documentation for additional information.

Non-Sticky Connections

A connection is called sticky if all packets are handled by a single cluster member. In a non-sticky connection, the reply packet returns via a different gateway than the original packet.

The synchronization mechanism knows how to properly handle non-sticky connections. In a non-sticky connection, a cluster member gateway can receive an out-of-state packet, which Security Gateway normally drops because it poses a security risk.

In Load Sharing configurations, all cluster members are active, and in Static NAT and encrypted connections, the source and destination IP addresses change. Therefore, Static NAT and encrypted connections through a Load Sharing cluster may be non-sticky. Non-stickiness may also occur with Hide NAT, but ClusterXL has a mechanism to make it sticky.

In High Availability configurations, all packets reach the Active machine, so all connections are sticky. If failover occurs during connection establishment, the connection is lost, but synchronization can be performed later.

If the other members do not know about a non-sticky connection, the packet will be out-of-state, and the connection will be dropped for security reasons. However, the Synchronization mechanism knows how to inform other members of the connection. The Synchronization mechanism thereby prevents out-of-state packets in valid, but non-sticky connections, so that these non-sticky connections are allowed.

Non-sticky connections will also occur if the network administrator has configured asymmetric routing, where a reply packet returns through a different gateway than the original packet.
Non-Sticky Connection Example: TCP 3-Way Handshake

The 3-way handshake that initiates all TCP connections can very commonly lead to a non-sticky (often called asymmetric routing) connection. The following situation may arise as depicted in this illustration:

Client A initiates a connection by sending a SYN packet to server B. The SYN passes through Gateway C, but the SYN/ACK reply returns through Gateway D. This is a non-sticky connection, because the reply packet returns through a different gateway than the original packet.

The synchronization network notifies Gateway D. If gateway D is updated before the SYN/ACK packet sent by server B reaches it, the connection is handled normally. If, however, synchronization is delayed, and the SYN/ACK packet is received on gateway D before the SYN flag has been updated, then the gateway will treat the SYN/ACK packet as out-of-state, and will drop the connection.

You can configure enhanced 3-Way TCP Handshake Enforcement to address this issue.

Synchronizing Non-Sticky Connections

The synchronization mechanism prevents out-of-state packets in valid, but non-sticky connections. The way it does this is best illustrated with reference to the 3-way handshake that initiates all TCP data connections. The 3-way handshake proceeds as follows:

1. SYN (client to server)
2. SYN/ACK (server to client)
3. ACK (client to server)
4. Data (client to server)

To prevent out-of-state packets, the following sequence (called "Flush and Ack") occurs:

1. Cluster member receives first packet (SYN) of a connection.
2. Suspects that it is non-sticky.
3. Hold the SYN packet.
4. Send the pending synchronization updates to all cluster members (including all changes relating to this packet).
5. Wait for all the other cluster members to acknowledge the information in the sync packet.
7. All cluster members are ready for the SYN-ACK.
Synchronizing Clusters on a Wide Area Network

Organizations are sometimes faced with the need to locate cluster members in geographical locations that are distant from each other. A typical example is a replicated data center whose locations are widely separated for disaster recovery purposes. In such a configuration it is clearly impractical to use a cross cable as the synchronization network.

The synchronization network can be spread over remote sites, which makes it easier to deploy geographically distributed clustering. There are two limitations to this capability:

1. The synchronization network must guarantee no more than 100ms latency and no more than 5% packet loss.
2. The synchronization network may only include switches and hubs. No routers are allowed on the synchronization network, because routers drop Cluster Control Protocol packets.

You can monitor and troubleshoot (see "Troubleshooting Synchronization" on page 64) geographically distributed clusters using the command line interface.

Synchronized Cluster Restrictions

The following restrictions apply to synchronizing cluster members:

- Only cluster members running on the identical platform can be synchronized.
- All cluster members must use the same Check Point software version.
- A user-authenticated connection through a cluster member will be lost if the cluster member goes down. Other synchronized cluster members will be unable to resume the connection.
- However, a client-authenticated connection or session-authenticated connection will not be lost.
- The reason for these restrictions is that user authentication state is maintained on Security Servers, which are processes, and thus cannot be synchronized on different machines in the way that kernel data can be synchronized. However, the state of session authentication and client authentication is stored in kernel tables, and thus can be synchronized.
- The state of connections using resources is maintained in a Security Server, so these connections cannot be synchronized for the same reason that user-authenticated connections cannot be synchronized.
- Accounting information is accumulated in each cluster member and reported separately to the Security Management server, where the information is aggregated. In case of a failover, accounting information that was accumulated on the failed member but not yet reported to the Security Management server is lost. To minimize the problem it is possible to reduce the period in which accounting information is "flushed". To do this, in the cluster object's Logs and Masters > Additional Logging page, configure the attribute Update Account Log every:

Configuring State Synchronization

Configure State synchronization as part of the process of configuring ClusterXL and OPSEC certified clustering products. Configuring State synchronization involves the following steps:

1. Setting up a synchronization network for the gateway cluster
2. Installing a Security Gateway and enabling synchronization during the configuration process
3. Enabling State Synchronization on the ClusterXL page for the cluster object

For configuration procedures, refer to the sections for configuring ClusterXL (see "Configuring ClusterXL" on page 36) and OPSEC certified cluster products (see "Configuring OPSEC Certified Clustering Products" on page 42).

Configuring a Service Not to Synchronize

To set a service not to synchronize:

1. In the Services branch of the objects tree, double click the TCP, UDP or Other type service that you do not wish to synchronize.
2. In the Service Properties window, click Advanced to display the Advanced Services Properties window.
3. Clear Synchronize connections on the cluster.

Creating Synchronized and Non-Synchronized Versions

It is possible to have both a synchronized and a non-synchronized definition of the service, and to use them selectively in the Security Rule Base.

1. Define a new TCP, UDP and Other type service. Give it a name that distinguishes it from the existing service.
2. Copy all the definitions from the existing service into the Service Properties window of the new service.
3. In the new service, click Advanced to display the Advanced Services Properties window.
4. Copy all the definitions from the existing service into the Advanced Service Properties window of the new service.
5. Set Synchronize connections on the cluster in the new service, so that it is different from the setting in the existing service.

Configuring Duration Limited Synchronization

Before you start this procedure, become familiar with the concept (see "Duration Limited Synchronization" on page 13).

Note - This feature is limited to HTTP-based services. The Start synchronizing option is not displayed for other services.

To configure duration limited synchronization:

1. In the Services branch of the objects tree, double click the TCP, UDP or Other type service that you wish to synchronize.
2. In the Service Properties window, click Advanced to display the Advanced Services Properties window.
3. Select Start synchronizing x seconds after connection initiation.
4. In the seconds field, enter the number of seconds or select the number of seconds from the list, for which you want synchronization to be delayed after connection initiation.
Chapter 3

Sticky Connections

In This Chapter

Introduction to Sticky Connections 17
The Sticky Decision Function 17
VPN Tunnels with 3rd Party Peers and Load Sharing 18
Third-Party Gateways in Hub and Spoke Deployments 18
Configuring the Sticky Decision Function 19
Establishing a Third-Party Gateway in a Hub and Spoke Deployment 20

Introduction to Sticky Connections

A connection is considered **sticky** when all of its packets are handled, in either direction, by a single cluster member. This is the case in High Availability mode, where all connections are routed through the same cluster member, and hence, sticky. This is also the case in Load Sharing mode when there are no VPN peers, static NAT rules or SIP.

In Load Sharing mode, however, there are cases where it is necessary to ensure that a connection that starts on a specific cluster member will continue to be processed by the same cluster member in both directions. To that end, certain connections can be made sticky by enabling the Sticky Decision Function.

**Note** - For the latest information regarding features that require sticky connections, refer to the R75.20 Release Notes (http://supportcontent.checkpoint.com/documentation_download?ID=12414).

The Sticky Decision Function

The Sticky Decision Function enables certain services to operate in a Load Sharing deployment. For example, it is required for L2TP traffic, or when the cluster is a participant in a site to site VPN tunnel with a third party peer.

The following services and connection types are now supported by enabling the Sticky Decision Function:

- VPN deployments with third-party VPN peers
- SecureClient/SecuRemote/SSL Network Extender encrypted connections, including SecureClient visitor mode

The Sticky Decision Function has the following limitations:

- Sticky Decision Function is not supported when employing either Performance Pack or a hardware-based accelerator card. Enabling the Sticky Decision Function disables these acceleration products.
- When the Sticky Decision Function is used in conjunction with VPN, cluster members are prevented from opening more than one connection to a specific peer. Opening another connection would cause another SA to be generated, which a third-party peer, in many cases, would not be able to process.
VPN Tunnels with 3rd Party Peers and Load Sharing

Check Point provides interoperability with third-party vendor gateways by enabling them to peer with Check Point gateways. A special case occurs when certain third-party peers (Microsoft LT2P, IPSO Symbian, and Cisco gateways and clients) attempt to establish VPN tunnels with ClusterXL Gateways in the Load Sharing mode. These peers are limited in their ability to store SAs, which means that a VPN session that begins on one cluster member and, due to load sharing, is routed on the return trip through another, is unrecognized and dropped.

The following illustration demonstrates this.

**Figure 3-3** Third-party peers connected to ClusterXL Load Sharing without Sticky Decision

In this scenario:
- A third-party peer (gateway or client) attempts to create a VPN tunnel.
- Cluster Members A and B belong to a ClusterXL Gateway in Load Sharing mode.

The third-party peers, lacking the ability to store more than one set of SAs, cannot negotiate a VPN tunnel with multiple cluster members, and therefore the cluster member cannot complete the routing transaction.

This issue is resolved for certain third-party peers or any gateways that can save only one set of SAs by making the connection sticky. Enabling the Sticky Decision Function sets all VPN sessions initiated by the same third-party gateway to be processed by a single cluster member.

**To enable the Sticky Decision Function:**
1. In SmartDashboard edit the cluster object > ClusterXL page > Advanced.
2. Enable the property **Use Sticky Decision Function**.

Third-Party Gateways in Hub and Spoke Deployments

Another case where Load Sharing mode requires the Sticky Decision Function is when integrating certain third-party gateways into a hub and spoke deployment. Without the ability to store more than one set of SAs, a third-party gateway must maintain its VPN tunnels on a single cluster member in order to avoid duplicate SAs.
Configuring the Sticky Decision Function

To configure the Sticky Decision Function:

1. Select a cluster object from the Network Object tree.
2. In the Advanced Load Sharing window, select one of the following options:

   - **IPs, Ports, SPIs** (default) provides the best sharing distribution, and is recommended for use. It is the least "sticky" sharing configuration.

In this scenario:

- The intent of this deployment is to enable hosts that reside behind Spoke A to communicate with hosts behind Spoke B.
- The ClusterXL Gateway is in Load Sharing mode, is composed of Cluster Members A and B, and serves as a VPN Hub.
- Spoke A is a third-party gateway, and is connected by a VPN tunnel that passes through the Hub to Spoke B.
- Spoke B can be either another third-party gateway or a Check Point Security Gateway.

Spokes A and B must be set to always communicate using the same cluster member. Enabling the Sticky Decision Function solves half of this problem, in that all VPN sessions initiated by either third-party gateway are processed by a single cluster member.

To make sure that all communications between Spokes A and B are always using the same cluster member, you must make some changes to the `user.def` file. This second step ensures that both third-party gateways always connect to the same cluster member (see "Establishing a Third-Party Gateway in a Hub and Spoke Deployment" on page 20).
IPs, Ports should be used only if problems arise when distributing IPSec packets to a few machines although they have the same source and destination IP addresses.

IPs should be used only if problems arise when distributing IPSec packets or different port packets to a few machines although they have the same source and destination IP addresses. It is the most "sticky" sharing configuration, in other words, it increases the probability that a certain

3. Enable the Sticky Decision Function option to enable its functionality or clear to disable. By default the Sticky Decision Function is disabled.

If enabled, the connection will pass through a single cluster member on both inbound and outbound directions.

Establishing a Third-Party Gateway in a Hub and Spoke Deployment

To establish a third-party gateway as a spoke in a hub and spoke deployment, perform the following on the Security Management server:

1. Enable the Sticky Decision Function if not already enabled. In SmartDashboard, edit the cluster object > ClusterXL page > Advanced, and enable the property Use Sticky Decision Function.

2. Create a Tunnel Group to handle traffic from specific peers. Use a text editor to edit the file $FWDIR/lib/user.def, and add a line similar to the following:

   all@{member1,member2} vpn_sticky_gws = {<10.10.10.1;1>,<20.20.20.1;1>};

The elements of this configuration are as follows:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>Stands for all the interfaces of the cluster Gateway</td>
</tr>
<tr>
<td>member1,member2</td>
<td>Names of the cluster members in SmartDashboard</td>
</tr>
<tr>
<td>vpn_sticky_gws</td>
<td>Name of the table</td>
</tr>
<tr>
<td>10.10.10.1</td>
<td>IP address of Spoke A</td>
</tr>
<tr>
<td>20.20.20.1</td>
<td>IP address of Spoke B</td>
</tr>
<tr>
<td>;1</td>
<td>Tunnel Group Identifier, which indicates that the traffic from these IP addresses should be handled by the same cluster member</td>
</tr>
</tbody>
</table>

3. Other peers can be added to the Tunnel Group by including their IP addresses in the same format as shown above. To continue with the example above, adding Spoke C would look like this:

   all@{member1,member2} vpn_sticky_gws = {<10.10.10.1;1>,<20.20.20.1;1>,<30.30.30.1;1>};

Note that the Tunnel Group Identifier ;1 stays the same, which means that the listed peers will always connect through the same cluster member.

Note - More tunnel groups than cluster members may be defined.

This procedure in essence turns off Load Sharing for the connections affected. If the implementation is to connect multiple sets of third-party gateways one to another, a form of Load Sharing can be accomplished by setting gateway pairs to work in tandem with specific cluster members. For instance, to set up a connection between two other spokes (C and D), simply add their IP addresses to the line and replace the Tunnel Group Identifier ;1 with ;2. The line would then look something like this:

   all@{member1,member2} vpn_sticky_gws = {<10.10.10.1;1>,<20.20.20.1;1>,<192.168.15.5;2>,<192.168.1.4;2>};
Note that there are now two peer identifiers: ;1 and ;2. Spokes A and B will now connect through one cluster member, and Spokes C and D through another.

**Note** - The tunnel groups are shared between active cluster members. In case of a change in cluster state (e.g., failover or member attach/detach), the reassignment is performed according to the new state.
Chapter 4

High Availability and Load Sharing in ClusterXL

Introduction to High Availability and Load Sharing

ClusterXL is a software-based Load Sharing and High Availability solution that distributes network traffic between clusters of redundant Security Gateways.

ClusterXL provides:

- Transparent failover in case of machine failures
- Zero downtime for mission-critical environments (when using State Synchronization)
- Enhanced throughput (in Load Sharing modes)
- Transparent upgrades

All machines in the cluster are aware of the connections passing through each of the other machines. The cluster members synchronize their connection and status information across a secure synchronization network.

The glue that binds the machines in a ClusterXL cluster is the Cluster Control Protocol (CCP), which is used to pass synchronization and other information between the cluster members.

Load Sharing

ClusterXL Load Sharing distributes traffic within a cluster of gateways so that the total throughput of multiple machines is increased.

In Load Sharing configurations, all functioning machines in the cluster are active, and handle network traffic (Active/Active operation).

If any individual Check Point gateway in the cluster becomes unreachable, transparent failover occurs to the remaining operational machines in the cluster, thus providing High Availability. All connections are shared between the remaining gateways without interruption.
High Availability

High Availability allows organizations to maintain a connection when there is a failure in a cluster member, without Load Sharing between cluster members. In a High Availability cluster, only one machine is active (Active/Standby operation). In the event that the active cluster member becomes unreachable, all connections are re-directed to a designated standby without interruption. In a synchronized cluster, the standby cluster members are updated with the state of the connections of the active cluster member.

In a High Availability cluster, each machine is given a priority. The highest priority machine serves as the gateway in normal circumstances. If this machine fails, control is passed to the next highest priority machine. If that machine fails, control is passed to the next machine, and so on.

On gateway recovery, you can maintain the current active gateway (Active Up), or to switch to the highest priority gateway (Primary Up).

In Active Up configuration, changing and installing the Security Policy can lead to a fail-over between the members. Another member can be chosen as the Active machine. During policy installation, the member initializes the ClusterXL configuration again, and sees if there is already an Active member in the cluster. If there is none, it becomes Active. This process is ClusterXL configuration handshake on the members.

Example ClusterXL Topology

ClusterXL uses unique physical IP and MAC addresses for each cluster member, and a virtual IP addresses for the cluster itself. Cluster interface addresses do not belong to any real machine interface.

The following diagram illustrates a two-member ClusterXL cluster, showing the cluster virtual IP addresses and member physical IP addresses. This sample deployment is used in many of the examples presented in this chapter.

Figure 4-5 Example ClusterXL Topology

Each cluster member has three interfaces: one external interface, one internal interface, and one for synchronization. Cluster member interfaces facing in each direction are connected via a switch, router, or VLAN switch.

All cluster member interfaces facing the same direction must be in the same network. For example, there must not be a router between cluster members.

The Security Management Server can be located anywhere, and should be routable to either the internal or external cluster addresses.

The following sections present ClusterXL configuration concepts shown in the example.
Defining the Cluster Member IP Addresses

The guidelines for configuring each cluster member machine are as follows:
All machines within the cluster must have at least three interfaces:

- An interface facing the external cluster interface, which in turn faces the internet.
- An interface facing the internal cluster interface, which in turn faces the internal network.
- An interface to use for synchronization.

All interfaces pointing in a certain direction must be on the same network.
For example, in the previous illustration, there are two cluster members, Member_A and Member_B. Each has an interface with an IP address facing the Internet through a hub or a switch. This is the External interface with IP address 192.168.10.1 on Member_A and 192.168.10.2 on Member_B, and is the interface that the cluster external interface sees.

Note - This release presents an option to use only two interfaces per member, one external and one internal and to run synchronization over the internal interface. This configuration is not recommended and should be used for backup only (see "Synchronizing Connection Information Across the Cluster" on page 11).

Defining the Cluster Virtual IP Addresses

In the previous illustration, the IP address of the cluster is 192.168.10.100.
The cluster has one external virtual IP address and one internal virtual IP address. The external IP address is 192.168.10.100, and the internal IP address is 10.10.0.100.

The Synchronization Network

State Synchronization between cluster members ensures that if there is a failover, connections that were handled by the failed machine will be maintained. The synchronization network is used to pass connection synchronization and other state information between cluster members. This network therefore carries all the most sensitive security policy information in the organization, and so it is important to make sure the network is secure. It is possible to define more than one synchronization network for backup purposes.

To secure the synchronization interfaces, they should be directly connected by a cross cable, or in a cluster with three or more members, use a dedicated hub or switch.

Machines in a Load Sharing cluster must be synchronized because synchronization is used in normal traffic flow. Machines in a High Availability cluster do not have to be synchronized, though if they are not, connections may be lost upon failover.

The previous illustration shows a synchronization interface with a unique IP address on each machine. 10.0.10.1 on Member_A and 10.0.10.2 on Member_B.
Configuring Cluster Addresses on Different Subnets

Only one routable IP address is required in a ClusterXL cluster, for the virtual cluster interface that faces the Internet. All cluster member physical IP addresses can be non-routable.

Configuring different subnets for the cluster IP addresses and the member addresses is useful in order to:

- Enable a multi-machine cluster to replace a single-machine gateway in a pre-configured network, without the need to allocate new addresses to the cluster members.
- Allow organizations to use only one routable address for the ClusterXL Gateway Cluster. This saves routable addresses.

ClusterXL Modes

ClusterXL has four working modes. This section briefly describes each mode and its relative advantages and disadvantages.

- Load Sharing Multicast Mode
- Load Sharing Unicast Mode
- New High Availability Mode
- High Availability Legacy Mode

Refer to the High Availability Legacy appendix (see "High Availability Legacy Mode" on page 106) for a detailed discussion of legacy High Availability functionality. It is recommended that you use the High Availability New Mode to avoid problems with backward compatibility.

**Note** - Many examples in the section refer to the sample deployment shown in the ClusterXL example ("Example ClusterXL Topology" on page 23).

Load Sharing Multicast Mode

Load Sharing enables you to distribute network traffic between cluster members. In contrast to High Availability, where only a single member is active at any given time, all cluster members in a Load Sharing solution are active, and the cluster is responsible for assigning a portion of the traffic to each member. This assignment is the task of a decision function, which examines each packet going through the cluster, and determines which member should handle it. Thus, a Load Sharing cluster utilizes all cluster members, which usually leads to an increase in its total throughput.

It is important to understand that ClusterXL Load Sharing, when combined with State Synchronization, provides a full High Availability solution as well. When all cluster members are active, traffic is evenly distributed between the machines. In case of a failover event, caused by a problem in one of the members, the processing of all connections handled by the faulty machine is immediately taken over by the other members.

ClusterXL offers two separate Load Sharing solutions: Multicast and Unicast (see "Load Sharing Unicast Mode" on page 26). The two modes differ in the way members receive the packets sent to the cluster. This section describes the Multicast mode.

The Multicast mechanism, which is provided by the Ethernet network layer, allows several interfaces to be associated with a single physical (MAC) address. Unlike Broadcast, which binds all interfaces in the same subnet to a single address, Multicast enables grouping within networks. This means that it is possible to select the interfaces within a single subnet that will receive packets sent to a given MAC address.

ClusterXL uses the Multicast mechanism to associate the virtual cluster IP addresses with all cluster members. By binding these IP addresses to a Multicast MAC address, it ensures that all packets sent to the cluster, acting as a gateway, will reach all members in the cluster. Each member then decides whether it should process the packets or not. This decision is the core of the Load Sharing mechanism: it has to assure that at least one member will process each packet (so that traffic is not blocked), and that no two members will handle the same packets (so that traffic is not duplicated).

An additional requirement of the decision function is to route each connection through a single gateway, to ensure that packets that belong to a single connection will be processed by the same member.
ClusterXL Modes

Unfortunately, this requirement cannot always be enforced, and in some cases, packets of the same connection will be handled by different members. ClusterXL handles these situations using its State Synchronization mechanism, which mirrors connections on all cluster members.

Example

This scenario describes a user logging from the Internet to a Web server behind the Firewall cluster that is configured in Load Sharing Multicast mode.

1. The user requests a connection from 192.168.10.78 (his computer) to 10.10.0.34 (the Web server).
2. A router on the 192.168.10.x network recognizes 192.168.10.100 (the cluster's virtual IP address) as the gateway to the 10.10.0.x network.
3. The router issues an ARP request to 192.168.10.100.
4. One of the active members intercepts the ARP request, and responds with the Multicast MAC assigned to the cluster IP address of 192.168.10.100.
5. When the Web server responds to the user requests, it recognizes 10.10.0.100 as its gateway to the Internet.
6. The Web server issues an ARP request to 10.10.0.100.
7. One of the active members intercepts the ARP request, and responds with the Multicast MAC address assigned to the cluster IP address of 10.10.0.100.
8. All packets sent between the user and the Web server reach every cluster member, which decides whether to handle or drop each packet.
9. When a failover occurs, one of the cluster members goes down. However, traffic still reaches all of the active cluster members, and hence there is no need to make changes in the network's ARP routing. All that changes is the cluster's decision function, which takes into account the new state of the members.

Load Sharing Unicast Mode

Load Sharing Unicast mode provides a Load Sharing solution adapted to environments where Multicast Ethernet cannot operate. In this mode a single cluster member, referred to as Pivot, is associated with the cluster's virtual IP addresses, and is thus the only member to receive packets sent to the cluster. The pivot is then responsible for propagating the packets to other cluster members, creating a Load Sharing mechanism. Distribution is performed by applying a decision function on each packet, the same way it is done in Load Sharing Multicast mode. The difference is that only one member performs this selection: any non-pivot member that receives a forwarded packet will handle it, without applying the decision function. Note that non-pivot members are still considered as "active", since they perform routing and Firewall tasks on a share of the traffic (although they do not perform decisions.).

Even though the pivot member is responsible for the decision process, it still acts as a Security Gateway that processes packets (for example, the decision it makes can be to handle a packet on the local machine). However, since its additional tasks can be time consuming, it is usually assigned a smaller share of the total load.

When a failover event occurs in a non-pivot member, its handled connections are redistributed between active cluster members, providing the same High Availability capabilities of New High Availability and Load Sharing Multicast. When the pivot member encounters a problem, a regular failover event occurs, and, in addition, another member assumes the role of the new pivot. The pivot member is always the active member with the highest priority. This means that when a former pivot recuperates, it will retain its previous role.

Example

In this scenario, we use a Load Sharing Unicast cluster as the gateway between the user's computer and the Web server.

1. The user requests a connection from 192.168.10.78 (his computer) to 10.10.0.34 (the Web server).
2. A router on the 192.168.10.x network recognizes 192.168.10.100 (the cluster's virtual IP address) as the gateway to the 10.10.0.x network.
3. The router issues an ARP request to 192.168.10.100.
4. The pivot member intercepts the ARP request, and responds with the MAC address that corresponds to its own unique IP address of 192.168.10.1.
5. When the Web server responds to the user requests, it recognizes 10.10.0.100 as its gateway to the Internet.
6. The Web server issues an ARP request to 10.10.0.100.
7. The pivot member intercepts the ARP request, and responds with the MAC address that corresponds to its own unique IP address of 10.10.0.1.
8. The user's request packet reaches the pivot member on interface 192.168.10.1.
9. The pivot decides that the second member should handle this packet, and forwards it to 192.168.10.2.
10. The second member recognizes the packet as a forwarded one, and processes it.
11. Further packets are processed by either the pivot member, or forwarded and processed by the non-pivot member.
12. When a failover occurs on the pivot, the second member assumes the role of pivot.
13. The new pivot member sends gratuitous ARP requests to both the 192.168.10.x and the 10.10.0.x networks. These requests associate the virtual IP address of 192.168.10.100 with the MAC address that corresponds to the unique IP address of 192.168.10.2, and the virtual IP address of 10.10.0.100 with the MAC address that correspond to the unique IP address of 10.10.0.2.
14. Traffic sent to the cluster is now received by the new pivot, and processed by the local machine (as it is currently the only active machine in the cluster).
15. When the first machine recovers, it re-assumes the role of pivot, by associating the cluster IP addresses with its own unique MAC addresses.

High Availability Mode

The High Availability Mode provides basic High-Availability capabilities in a cluster environment. This means that the cluster can provide Firewall services even when it encounters a problem, which on a stand-alone gateway would have resulted in a complete loss of connectivity. When combined with Check Point's State Synchronization, ClusterXL High Availability can maintain connections through failover events, in a user-transparent manner, allowing a flawless connectivity experience. Thus, High-Availability provides a backup mechanism, which organizations can use to reduce the risk of unexpected downtime, especially in a mission-critical environment (such as one involving money transactions over the Internet.)

To achieve this purpose, ClusterXL's New High Availability mode designates one of the cluster members as the active machine, while the other members remain in stand-by mode. The cluster's virtual IP addresses are associated with the physical network interfaces of the active machine (by matching the virtual IP address with the unique MAC address of the appropriate interface). Thus, all traffic directed at the cluster is actually routed (and filtered) by the active member. The role of each cluster member is chosen according to its priority, with the active member being the one with the highest ranking. Member priorities correspond to the order in which they appear in the Cluster Members page of the Gateway Cluster Properties window. The top-most member has the highest priority. You can modify this ranking at any time.

In addition to its role as a Firewall gateway, the active member is also responsible for informing the stand-by members of any changes to its connection and state tables, keeping these members up-to-date with the current traffic passing through the cluster.

Whenever the cluster detects a problem in the active member that is severe enough to cause a failover event, it passes the role of the active member to one of the standby machines (the member with the currently highest priority). If State Synchronization is applied, any open connections are recognized by the new active machine, and are handled according to their last known state. Upon the recovery of a member with a higher priority, the role of the active machine may or may not be switched back to that member, depending on the user's configuration.

It is important to note that the cluster may encounter problems in standby machines as well. In this case, these machines are not considered for the role of active members, in the event of a failover.

Example

This scenario describes a user logging from the Internet to a Web server behind the Firewall cluster.
1. The user requests a connection from 192.168.10.78 (his computer) to 10.10.0.34 (the Web server).
2. A router on the 192.168.10.x network recognizes 192.168.10.100 (the cluster's virtual IP address) as the gateway to the 10.10.0.x network.
3. The router issues an ARP request to 192.168.10.100.
4. The active member intercepts the ARP request, and responds with the MAC address that corresponds to its own unique IP address of 192.168.10.1.

5. When the Web server responds to the user requests, it recognizes 10.10.0.100 as its gateway to the Internet.

6. The Web server issues an ARP request to 10.10.0.100.

7. The active member intercepts the ARP request, and responds with the MAC address that corresponds to its own unique IP address of 10.10.0.1.

8. All traffic between the user and the Web server is now routed through the active member.

9. When a failover occurs, the standby member concludes that it should now replace the faulty active member.

10. The stand-by member sends gratuitous ARP requests to both the 192.168.10.x and the 10.10.0.x networks. These requests associate the virtual IP address of 192.168.10.100 with the MAC address that corresponds to the unique IP address of 192.168.10.2, and the virtual IP address of 10.10.0.100 with the MAC address that correspond to the unique IP address of 10.10.0.2.

11. The stand-by member has now switched to the role of the active member, and all traffic directed through the cluster is routed through this machine.

12. The former active member is now considered to be “down”, waiting to recover from whatever problem that had caused the failover event.

Mode Comparison Table
This table summarizes the similarities and differences between the ClusterXL modes.

<table>
<thead>
<tr>
<th></th>
<th>Legacy High Availability</th>
<th>New High Availability</th>
<th>Load Sharing Multicast</th>
<th>Load Sharing Unicast</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Availability</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Load Sharing</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Performance</td>
<td>Good</td>
<td>Good</td>
<td>Excellent</td>
<td>Very Good</td>
</tr>
<tr>
<td>Hardware Support</td>
<td>All</td>
<td>All</td>
<td>Not all routers are supported</td>
<td>All</td>
</tr>
<tr>
<td>SecureXL Support</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes, with Performance Pack or SecureXL Turbocard.</td>
<td>Yes</td>
</tr>
<tr>
<td>State Synchronization Mandatory</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>VLAN Tagging Support</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Note - For further details regarding VLAN Tagging Support, refer to the R75.20 Release Notes (http://supportcontent.checkpoint.com/documentation_download?ID=12414).
Failover

A failover occurs when a Gateway is no longer able to perform its designated functions. When this happens another Gateway in the cluster assumes the failed Gateway's responsibilities.

In a Load Sharing configuration, if one Security Gateway in a cluster of gateways goes down, its connections are distributed among the remaining Gateways. All gateways in a Load Sharing configuration are synchronized, so no connections are interrupted.

In a High Availability configuration, if one Gateway in a synchronized cluster goes down, another Gateway becomes active and "takes over" the connections of the failed Gateway. If you do not use State Synchronization, existing connections are closed when failover occurs, although new connections can be opened.

To tell each cluster member that the other gateways are alive and functioning, the ClusterXL Cluster Control Protocol maintains a heart beat between cluster members. If a certain predetermined time has elapsed and no message is received from a cluster member, it is assumed that the cluster member is down and a failover occurs. At this point another cluster member automatically assumes the responsibilities of the failed cluster member.

It should be noted that a cluster machine may still be operational but if any of the above checks fail in the cluster, then the faulty member initiates the failover because it has determined that it can no longer function as a cluster member.

Note that more than one cluster member may encounter a problem that will result in a failover event. In cases where all cluster members encounter such problems, ClusterXL will try to choose a single member to continue operating. The state of the chosen member will be reported as Active Attention. This situation lasts until another member fully recovers. For example, if a cross cable connecting the cluster members malfunctions, both members will detect an interface problem. One of them will change to the Down state, and the other to Active Attention.

When Does a Failover Occur?

A failover takes place when one of the following occurs on the active cluster member:

- Any critical device (such as fwd) fails. 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 failover has occurred and another cluster member takes over.
- An interface or cable fails.
- The machine crashes.
- The Security Policy is uninstalled. When the Security Policy is uninstalled the Gateway can no longer function as a firewall. If it cannot function as a firewall, it can no longer function as a cluster member and a failover occurs. Normally a policy is not uninstalled by itself but would be initiated by a user. See sk36320 (http://supportcontent.checkpoint.com/solutions?id=sk36320) for more information.

What Happens When a Gateway Recovers?

In a Load Sharing configuration, when the failed Gateway in a cluster recovers, all connections are redistributed among all active members.

In a High Availability configuration, when the failed Gateway in a cluster recovers, the recovery method depends on the configured cluster setting. The options are:

- **Maintain Current Active Gateway** means that if one machine passes on control to a lower priority machine, control will be returned to the higher priority machine only if the lower priority machine fails. This mode is recommended if all members are equally capable of processing traffic, in order to minimize the number of failover events.
- **Switch to Higher Priority Gateway** means that if the lower priority machine has control and the higher priority machine is restored, then control will be returned to the higher priority machine. This mode is recommended if one member is better equipped for handling connections, so it will be the default gateway.
How a Recovered Cluster Member Obtains the Security Policy

The administrator installs the security policy on the cluster rather than separately on individual cluster members. The policy is automatically installed on all cluster members. The policy is sent to the IP address defined in the General Properties page of the cluster member object.

When a failed cluster member recovers, it will first try to take a policy from one of the other cluster members. The assumption is that the other cluster members have a more up to date policy. If this does not succeed, it compares its own local policy to the policy on the Security Management server. If the policy on the Security Management server is more up to date than the one on the cluster member, the policy on the Security Management server will be retrieved. If the cluster member does not have a local policy, it retrieves one from the Security Management server. This ensures that all cluster members use the same policy at any given moment.

Implementation Planning Considerations

High Availability or Load Sharing

Whether to choose a Load Sharing (Active/Active) or a High Availability (Active/Standby) configuration depends on the need and requirements of the organization. A High Availability gateway cluster ensures fail-safe connectivity for the organization. Load Sharing provides the additional benefit of increasing performance.

Note - When working on a sync network, it is recommended to use a NIC with the same bandwidth as the NICs that are used for general traffic.

Choosing the Load Sharing Mode

Load Sharing Multicast mode is an efficient way to handle a high load because the load is distributed optimally between all cluster members. However, not all routers can be used for Load Sharing Multicast mode. Load Sharing Multicast mode associates a multicast MAC with each unicast cluster IP address. This ensures that traffic destined for the cluster is received by all members. The ARP replies sent by a cluster member will therefore indicate that the cluster IP address is reachable via a multicast MAC address.

Some routing devices will not accept such ARP replies. For some routers, adding a static ARP entry for the cluster IP address on the routing device will solve the issue. Other routers will not accept this type of static ARP entry.

Another consideration is whether your deployment includes routing devices with interfaces operating in promiscuous mode. If on the same network segment there exists two such routers and a ClusterXL gateway in Load Sharing Multicast mode, traffic destined for the cluster that is generated by one of the routers could also be processed by the other router.

For these cases, use Load Sharing Unicast mode, which does not require the use of multicast for the cluster addresses.

IP Address Migration

If you wish to provide High Availability or Load Sharing to an existing single gateway configuration, it is recommended to take the existing IP addresses from the current gateway, and make these the cluster addresses (cluster virtual addresses), when feasible. Doing so will avoid altering current IPSec endpoint identities, as well keep Hide NAT configurations the same in many cases.
Hardware Requirements, Compatibility and Cisco Example

ClusterXL Hardware Requirements

The Gateway Cluster is usually located in an environment having other networking devices such as switches and routers. These devices and the Gateways must interact to assure network connectivity. This section outlines the requirements imposed by ClusterXL on surrounding networking equipment.

HA New and Load Sharing Unicast Modes

Multicast mode is the default Cluster Control Protocol (CCP) mode in High Availability New Mode and Load Sharing Unicast Mode (and also Load Sharing Multicast Mode).

When using CCP in multicast mode, configure the following settings on the switch.

Table 4-2 Switch Setting for High Availability New Mode and Load Sharing

<table>
<thead>
<tr>
<th>Switch Setting</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>IGMP and Static CAMs</td>
<td>ClusterXL does not support IGMP registration (also known as IGMP Snooping) by default. Either disable IGMP registration in switches that rely on IGMP packets to configure their ports, or enable IGMP registration on ClusterXL. In situations where disabling IGMP registration is not acceptable, it is necessary to configure static CAMs in order to allow multicast traffic on specific ports.</td>
</tr>
<tr>
<td>Disabling multicast limits</td>
<td>Certain switches have an upper limit on the number of broadcasts and multicasts that they can pass, in order to prevent broadcast storms. This limit is usually a percentage of the total interface bandwidth. If the connecting switch is incapable of having any of these settings configured, it is possible, though less efficient, for the switch to use broadcast to forward traffic, and to configure the cluster members to use broadcast CCP (see &quot;Choosing the CCP Transport Mode on the Cluster Members&quot; on page 37).</td>
</tr>
</tbody>
</table>

Configure the following settings on the router:

Table 4-3 Router Setting for High Availability New Mode and Load Sharing Unicast Mode

<table>
<thead>
<tr>
<th>Router Setting</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unicast MAC</td>
<td>When working in High Availability Legacy mode, High Availability New mode and Load Sharing Unicast mode, the Cluster IP address is mapped to a regular MAC address, which is the MAC address of the active member. The router needs to be able to learn this MAC through regular ARP messages.</td>
</tr>
</tbody>
</table>
Load Sharing Multicast Mode

When working in Load Sharing Multicast mode, the switch settings are as follows:

Table 4-4  Switch Configuration for Load Sharing Multicast Mode

<table>
<thead>
<tr>
<th>Switch Setting</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCP in Multicast mode</td>
<td>Multicast mode is the default Cluster Control Protocol mode in Load Sharing Multicast. For details of the required switch settings, see Table 4-2 on page 70.</td>
</tr>
<tr>
<td>Port Mirroring</td>
<td>ClusterXL does not support the use of unicast MAC addresses with Port Mirroring for Multicast Load Sharing solutions.</td>
</tr>
</tbody>
</table>

When working in Load Sharing Multicast mode, the router must support sending unicast IP packets with Multicast MAC addresses. This is required so that all cluster members will receive the data packets.

The following settings may need to be configured in order to support this mode, depending on the model of the router:

Table 4-5  Router Configuration for Load Sharing Multicast Mode

<table>
<thead>
<tr>
<th>Router Setting</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static MAC</td>
<td>Most routers can learn ARP entries with a unicast IP and a multicast MAC automatically using the ARP mechanism. If you have a router that is not able to learn this type of mapping dynamically, you'll have to configure static MAC entries.</td>
</tr>
<tr>
<td>IGMP and static cams</td>
<td>Some routers require disabling of IGMP snooping or configuration of static cams in order to support sending unicast IP packets with Multicast MAC addresses.</td>
</tr>
<tr>
<td>Disabling multicast limits</td>
<td>Certain routers have an upper limit on the number of broadcasts and multicasts that they can pass, in order to prevent broadcast storms. This limit is usually a percentage of the total interface bandwidth. It is possible to either turn off broadcast storm control, or to allow a higher level of broadcasts or multicasts through the router.</td>
</tr>
<tr>
<td>Disabling forwarding multicast traffic to the router</td>
<td>Some routers will send multicast traffic to the router itself. This may cause a packet storm through the network and should be disabled.</td>
</tr>
</tbody>
</table>

ClusterXL Hardware Compatibility

The following routers and switches are known to be compatible for all ClusterXL modes:

Routers

- Cisco 7200 Series
- Cisco 1600, 2600, 3600 Series

Routing Switch

- Extreme Networks Blackdiamond (Disable IGMP snooping)
- Extreme Networks Alpine 3800 Series (Disable IGMP snooping)
- Foundry Network Bigiron 4000 Series
- Nortel Networks Passport 8600 Series
• Cisco Catalyst 6500 Series (Disable IGMP snooping, Configure Multicast MAC manually)

**Switches**

• Cisco Catalyst 2900, 3500 Series
• Nortel BayStack 450
• Alteon 180e
• Dell PowerConnect 3248 and PowerConnect 5224

**Example Configuration of a Cisco Catalyst Routing Switch**

The following example shows how to perform the configuration commands needed to support ClusterXL on a Cisco Catalyst 6500 Series routing switch. For more details, or instructions for other networking devices, please refer to the device vendor documentation.

**Disabling IGMP Snooping**

To disable IGMP snooping run:

```
no ip igmp snooping
```

**Defining Static Cam Entries**

To add a permanent multicast entry to the table for module 1, port 1, and module 2, ports 1, 3, and 8 through 12:

1. Console> (enable) set cam permanent 01-40-5e-28-0a-64 1/1,2/1,2/3,2/8-12
2. Permanent multicast entry added to CAM table.
3. Console> (enable)

To determine the MAC addresses which needs to be set:

1. On a network that has a cluster IP address of x.y.z.w:
   - If y<=127, the multicast MAC address would be 01:00:5e:y:z:w. For example: 01:00:5e:5A:0A:64 for 192.90.10.100
   - If y>127, the multicast MAC address would be 01:00:5e:(y-128):z:w. For example: 01:00:5e:28:0A:64 for 192.168.10.100 (168-128=40 = 28 in hex).
2. For a network x.y.z.0 that does not have a cluster IP address, such as the sync, you would use the same procedure, and substitute fa instead of 0 for the last octet of the MAC address.
   - For example: 01:00:5e:00:00:fa for the 10.0.0.X network.

**Disabling Multicast Limits**

To disable multicast limits run:

```
no storm-control multicast level
```

**Configuring a Static ARP Entry on the Router**

To define a static ARP entry:

1. Determine the MAC address (see "Defining Static Cam Entries" on page 33).
2. Run `arp <MAC address> arpa`

**Disabling Multicast Packets from Reaching the Router**

To prevent multicast packets from reaching the router:

1. Determine the MAC address (see "Defining Static Cam Entries" on page 33).
2. Run `set cam static <MAC address> module/port`. 
Check Point Software Compatibility

Operating System Compatibility

The operating systems listed in the table below are supported by ClusterXL, with the limitations listed in the notes below. For details on the supported versions of these operating systems, refer to the R75.20 Release Notes (http://supportcontent.checkpoint.com/documentation_download?ID=12414).

Table 4-6  *ClusterXL Operating System Compatibility*

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Load Sharing</th>
<th>High Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check Point SecurePlatform</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

*Notes*

1. VLANs are supported on all interfaces.

ClusterXL Compatibility (excluding IPS)

The following table and accompanying notes present Cluster XL Load Sharing and High Availability compatibility for OPSEC Certified cluster products (see “Working with OPSEC Certified Clustering Products” on page 42). Some Check Point products and features are not supported or are only partially supported (as detailed in the footnotes) for use with ClusterXL.

Table 4-7  *Products and features that are not fully supported with ClusterXL*

<table>
<thead>
<tr>
<th>Feature or Product</th>
<th>Feature</th>
<th>Load Sharing</th>
<th>High Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security Management</td>
<td></td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Firewall</td>
<td>Authentication/Security Servers</td>
<td>Yes (1.)</td>
<td>Yes (1.) (9.)</td>
</tr>
<tr>
<td>Firewall</td>
<td>ACE servers and SecurID</td>
<td>Yes (7.)</td>
<td>Yes (7.)</td>
</tr>
<tr>
<td>Firewall</td>
<td>Application Intelligence protocol inspection</td>
<td>Yes (3.)</td>
<td>Yes</td>
</tr>
<tr>
<td>Firewall</td>
<td>Sequence Verifier</td>
<td>Yes (4.)</td>
<td>Yes (1.)</td>
</tr>
<tr>
<td>Firewall</td>
<td>UDP encapsulation</td>
<td>Yes (7.)</td>
<td>Yes</td>
</tr>
<tr>
<td>Firewall</td>
<td>SAM</td>
<td>Yes (8.)</td>
<td>Yes (8.)</td>
</tr>
<tr>
<td>Firewall</td>
<td>ISP Redundancy</td>
<td>Yes (11.) (12.)</td>
<td>Yes (11.) (13.)</td>
</tr>
<tr>
<td>VPN</td>
<td>Third party VPN peers</td>
<td>Yes (16.)</td>
<td>Yes</td>
</tr>
<tr>
<td>&lt;esp&gt; Client</td>
<td>Software Distribution Server (SDS)</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>&lt;esp&gt; Client</td>
<td>IP per user in Office Mode</td>
<td>Yes (10.)</td>
<td>Yes (10.)</td>
</tr>
<tr>
<td>SecureXL (hardware acceleration) or Performance Pack</td>
<td>Yes (11.) (15.)</td>
<td>Yes (11.)</td>
<td></td>
</tr>
<tr>
<td>Check Point QoS</td>
<td></td>
<td>Yes (4.) (5.)</td>
<td>Yes</td>
</tr>
<tr>
<td>Feature or Product</td>
<td>Feature</td>
<td>Load Sharing</td>
<td>High Availability</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>----------------------------------</td>
<td>--------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>SmartProvisioning</td>
<td>SmartLSM Security Gateway</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Check Point Security Gateway</td>
<td></td>
<td>Yes (6.)</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**ClusterXL Compatibility with IPS**

The following IPS features are supported by ClusterXL, with the limitations listed in the notes.

*Table 4-8 ClusterXL Compatibility with IPS*

<table>
<thead>
<tr>
<th>Feature</th>
<th>Load Sharing</th>
<th>High Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fragment Sanity Check</td>
<td>Yes (1, 3)</td>
<td>Yes (1)</td>
</tr>
<tr>
<td>Pattern Matching</td>
<td>Yes (2, 3)</td>
<td>Yes (2)</td>
</tr>
<tr>
<td>Sequence Verifier</td>
<td>Yes (2, 4)</td>
<td>Yes (2)</td>
</tr>
<tr>
<td>FTP, HTTP and SMTP Security Servers</td>
<td>Yes (2, 5)</td>
<td>Yes (2)</td>
</tr>
</tbody>
</table>

**Footnotes**

1. If there is a failover when fragments are being received, the packet will be lost.
2. Does not survive failover.
3. Requires unidirectional stickiness. This means that the same member must receive all external packets, and the same member must receive all internal packets, but the same member does not have to receive both internal and external packets.
4. Requires bidirectional connection stickiness.
5. Uses the forwarding layer, described in the next section.

**Forwarding Layer**

The Forwarding Layer is a ClusterXL mechanism that allows a cluster member to pass packets to other members, after they have been locally inspected by the Firewall. This feature allows connections to be opened from a cluster member to an external host.

Packets originated by cluster members are hidden behind the cluster's virtual IP. Thus, a reply from an external host is sent to the cluster, and not directly to the source member. This can pose problems in the following situations:

- The cluster is working in New High Availability mode, and the connection is opened from the stand-by machine. All packets from the external host are handled by the active machine, instead.
- The cluster is working in a Load Sharing mode, and the decision function has selected another member to handle this connection. This can happen since packets directed at a cluster IP are distributed among cluster members as with any other connection.

If a member decides, upon the completion of the Firewall inspection process, that a packet is intended for another cluster member, it can use the Forwarding Layer to hand the packet over to that destination. This is done by sending the packet over a secured network (any subnet designated as a Synchronization network) directly to that member. It is important to use secured networks only, as encrypted packets are decrypted during the inspection process, and are forwarded as clear-text (unencrypted) data.

Packets sent on the Forwarding Layer use a special source MAC address to inform the receiving member that they have already been inspected by another Security Gateway. Thus, the receiving member can safely hand over these packets to the local Operating System, without further inspection. This process is secure, as Synchronization Networks should always be isolated from any other network (using a dedicated network).
Chapter 5

Configuring ClusterXL

This procedure describes how to configure the Load Sharing Multicast, Load Sharing Unicast, and High Availability New Modes modes from scratch. Their configuration is identical, apart from the mode selection in SmartDashboard Gateway Cluster object or Gateway Cluster creation wizard.

If you are still using the High Availability Legacy Mode, refer to the appendix (see "High Availability Legacy Mode" on page 106).

In This Chapter

Preparing the Cluster Member Machines 36
Configuring Routing for Client Machines 37
Choosing the CCP Transport Mode on the Cluster Members 37
Configuring Cluster Objects & Members 37

Preparing the Cluster Member Machines

To prepare cluster members:
1. Obtain and install a ClusterXL central license on your Security Management server.
2. Install and configure Check Point Security Gateway on all cluster members. Each member must use the identical version and build. Refer to the R75.20 Installation and Upgrade Guide (http://supportcontent.checkpoint.com/documentation_download?ID=12269) for installation and initial configuration procedures.
   a) During installation process, enable ClusterXL and State Synchronization by selecting Enable cluster membership for this gateway for Secure Platform and Solaris members, or This Gateway is part of a cluster for Windows-based members.
      If you did not perform this action during installation, you can always do so by using the cpconfig utility at a later time. Run the cpconfig from the command line, and select the appropriate options to enable cluster capabilities for that member. You may be asked to reboot the member.
3. Define an IP address for each interface on all members.
4. For VPN cluster members, synchronize member clocks accurately to within one second of each other. If these members are constantly up and running it is usually enough to set the time once. More reliable synchronization can be achieved using NTP or some other time synchronization services supplied by the operating system. Cluster member clock synchronization is not relevant for non VPN cluster functionality.
5. Connect the cluster members to each other and to the networks via switches. For the synchronization interfaces, you can use a cross cable or a dedicated switch. Make sure that each network (internal, external, Synchronization, DMZ, and so on) is configured on a separate VLAN, switch or hub.

Note - You can also perform synchronization over a WAN (see "Synchronizing Clusters on a Wide Area Network" on page 15).
Configuring Routing for Client Machines

To configure routing for the client machines:

1. Configure routing so that communication with internal networks uses the external cluster IP address. For example, configure a static route such that internal network 10.10.0.0 is reached via 192.168.10.100.
2. Configure routing so that communication with external networks uses the internal cluster IP address. For example, define the internal network IP address 10.10.0.100 as the default gateway for each computer on the internal side of the router.

Choosing the CCP Transport Mode on the Cluster Members

The ClusterXL Control Protocol (CCP) uses multicast by default, because it is more efficient than broadcast. If the connecting switch cannot forward multicast traffic, it is possible, though less efficient, for the switch to use broadcast to forward traffic.

To toggle the CCP mode between broadcast and multicast:

- On each cluster member run the following command from the command line:
  
  cphaconf set_ccp broadcast/multicast

Configuring Cluster Objects & Members

Use the Cluster object Topology page to configure the topology for the cluster object and its members. The cluster IP addresses are virtual, as they do not belong to any physical interface. Assign one or more synchronization network interfaces for each cluster member.

The following network diagram illustrates a typical ClusterXL deployment and is used in many of the examples presented with the procedures in this chapter.

Figure 5-6 Example ClusterXL Topology
To define a new Gateway Cluster object:
1. In the Network Objects tree, right-click Check Point and then select Security Cluster.
2. If the Security Gateway Cluster Creation window appears, select one of the following methods to create your new cluster object:
   - **Simple Mode (Wizard)**, which guides you step by step through the configuration process. Refer to the online help for further assistance.
   - **Classic Mode**, as described in the following section.

### Using the Wizard

#### Classic Mode Configuration

**Configuration Steps**

- Configuring General Properties
- Defining Cluster Members
- Configuring ClusterXL Properties
- Configuring the Cluster Topology
- Completing the Definition

### Configuring General Properties

On the cluster General Properties page, configure the following:

![Gateway Cluster Properties - Corporate-Cluster-2](image-url)
1. Enter a unique name for this cluster object in the designated field.
2. Enter the main cluster IP address. This should be one of the cluster virtual interface IP addresses.
3. Select ClusterXL as a product installed on the cluster.
4. Enable ClusterXL and other Network Security Software Blades as appropriate.

**Defining Cluster Members**

On the Cluster Members page, click Add > New Cluster Member and configure the following:

1. In the Cluster Members Properties window General tab, enter a member Name and a physical IP Address, which must be routable from the Security Management server. This can be an internal or an external address, or a dedicated management interface.
2. Click Communication, and initialize Secure Internal Communication (SIC) trust.

**Adding an Existing Gateway as a Member**

To add an existing gateway as a cluster member, select Add > Add Gateway to Cluster on the Cluster Members page and select a gateway from the list in the Add Gateway to Cluster window.

**Removing a Member**

To remove a gateway from the cluster, click Remove in the Cluster Members page and select Detach Member from Cluster or right-click on the cluster member in the Network Objects tree and select Detach from Cluster.
Configuring ClusterXL Properties

To Configure ClusterXL properties:

1. For High Availability deployments, enable the **High Availability** option and
   a) Select one of the following modes:
      - New (default)
      - Legacy
   b) Select the action to perform upon primary member recovery:
      - Maintain the current active cluster member
      - Switch back to a higher priority cluster member

2. For Load Sharing deployments, enable the **Load Sharing** option and select one of the following modes according to your router's capabilities:
   - Multicast (Default)
   - Unicast

3. Enable the **Use State Synchronization** option as indicated:
   a) Load Sharing configurations require State Synchronization. This option is enabled by default and you cannot disable it.
   b) For High Availability New deployments, State Synchronization is optional, but enabled by default. If you choose to disable State Synchronization, cluster members do not synchronize, and existing connections on the failed gateway will be terminate once failover occurs.

4. Select tracking options from the list.

**Configuring the Cluster Topology**

1. In the **Topology** page, click **Edit Topology** to define the virtual cluster IP addresses and at least one synchronization network.
2. In the **Edit Topology** window:

![Edit Topology Window](image)

a) Define the topology for each member interface. To automatically read all predefined settings on member interfaces, click **Copy topology to cluster interfaces**.

b) In the **Network Objective** column, select an objective for each network from the its list. The various options are explained in the online help. To define a new network, click **Add Network**.

3. Define the topology for each virtual cluster interface. In a virtual cluster interface cell, right click and select **Edit Interface**. The **Interface Properties** window opens.

a) In the **General** tab, **Name** the virtual interface, and define an **IP Address**.

b) In the **Topology** tab, define whether the interface is internal or external, and set up anti-spoofing.

c) In the **Member Networks** tab, define the member network and net mask if necessary. This advanced option is explained in Configuring Cluster Addresses on Different Subnets.

**Completing the Definition**

1. Define the other pages in the cluster object as required (**NAT, VPN, Remote Access**, and so on).

Chapter 6

Working with OPSEC Certified Clustering Products

Introduction to OPSEC Certified Clustering Products

There are a number of OPSEC certified High Availability (sometimes called as Hot Standby) and Load Sharing (sometimes called Load Balancing) products. These products are used to build highly available Security Gateway clusters and to distribute traffic evenly among the clustered gateways.

Each OPSEC certified clustering application has its particular strengths and capabilities, whether it be monitoring, management, or performance. The role of these clustering applications is to:

1. Decide which cluster member will deal with each connection.
2. Perform health checks. This involves checking the status of a cluster member (for example, Active, Standby, or Down), and checking the status of the member interfaces.
3. Perform failover.

OPSEC certified clustering products use Check Point state synchronization mechanism (described in Synchronizing Connection Information Across the Cluster (on page 11)) to exchange and update connection information and other states between cluster members.

This guide provides general guidelines for working with OPSEC certified clustering products. Configuration details vary for each clustering product. You are therefore urged to follow the instructions supplied with the OPSEC product.

Configuring OPSEC Certified Clustering Products

This procedure describes how to configure an OPSEC certified Security Gateway clustering solution.

Preparing the Switches and Configuring Routing

Follow the instructions in your clustering product documentation for:

- Preparing the switches and routers
- Configuring routing
Preparing the Cluster Member Machines

To prepare the cluster member machines:
1. Define IP addresses for all interfaces on all the cluster members.
2. Connect the cluster network machines, via the switches. For the Synchronization interfaces, a cross-over cable or a dedicated switch is recommended.

   Note - It is possible to run synchronization across a WAN. For details, see Synchronizing Clusters over a Wide Area Network (see "Synchronizing Clusters on a Wide Area Network" on page 15).

3. For IPSO clusters, configure VRRP or IP Clustering before installing Check Point Security Gateway.
4. For OPSEC certified clusters, follow the vendor recommendations.

After the installation has finished, make sure that the Enable VPN-1/FW-1 monitoring is set to Enable in the IPSO configuration manager. This assures that IPSO will monitor changes in the status of the firewall. For VRRP and IP Clustering in IPSO 3.8.2 and above, the state of the firewall is reported to the IPSO cluster for failover purposes.
5. Install Check Point Security Gateway on all cluster members. During the configuration phase (or later, using the cpconfig Configuration Tool):
   - Install a license for Check Point Security Gateway on each cluster member. No special license is required to allow the OPSEC certified product to work with the Security Gateway.
   - During the configuration phase, enable State Synchronization by selecting Enable cluster membership for this gateway on Unix machines, or This Gateway is part of a cluster on Windows.

SmartDashboard Configuration for OPSEC Clusters

Using SmartDashboard, create the Gateway Cluster object. To define a new Gateway Cluster object, right click the Network Objects tree, and choose New Check Point > Gateway Cluster.... Configuration of the Gateway Cluster Object can be performed using:

- **Simple Mode (Wizard)** which guides you step by step through the configuration process. See the online help for further assistance.
- **Classic Mode**, described below.

**Classic Mode Configuration**

To use Classic Mode configuration with OPSEC:
1. In the General Properties page of the Gateway Cluster object, give the cluster a general IP address. In general, make it the external virtual IP address of the cluster.
   In the list of Check Point Products, ensure ClusterXL is not selected.
2. In the Cluster Members page, click Add > New Cluster Member to add cluster members to the cluster.
   Cluster members exist solely inside the Gateway Cluster object. For each cluster member:
   - In the Cluster Members Properties > General tab, define a name a Name and IP Address.
     Choose an IP address that is routable from the Security Management server so that the Security Policy installation will be successful. This can be an internal or an external address, or a dedicated management interface.
   - Click Communication, and Initialize Secure Internal Communication (SIC).
   - Define the NAT and VPN tabs, as required.
   You can also add an existing gateway as a cluster member by selecting Add > Add Gateway to Cluster in the Cluster Members page and selecting the gateway from the list in the Add Gateway to Cluster window.
   If you want to remove a gateway from the cluster, click Remove in the Cluster Members page and select Detach Member from Cluster or right-click on the cluster member in the Network Objects tree and select Detach from Cluster.
3. In the 3rd Party Configuration page, specify the cluster operating mode, and for the 3rd Party Solution, select OPSEC, and check Use State Synchronization.
4. The Topology page is used to define the virtual cluster IP addresses and cluster member addresses.
For each cluster member, define the interfaces for the individual members. For OPSEC certified products, the configuration of virtual cluster IPs is mandatory in several products, while in others it is forbidden. Refer to your cluster product documentation for details.

Define the synchronization networks. Depending on the OPSEC implementation, it might be possible to "get" the synchronization network from the OPSEC configuration if it is already defined. Refer to the OPSEC documentation to find out if this feature is implemented for a specific OPSEC product.

5. Now go back to the 3rd Party Configuration page.

A non-sticky connection is one in which packets from client to server and from server to client pass through different cluster members. Non-sticky connections are a problem because they can lead to out-of-state packets being received by the cluster member. The Security Gateway will reject out-of-state packets, even if they belong to a valid connection.

Either the synchronization mechanism, or the OPSEC certified clustering product need to be able to identify valid non-sticky connections, so that the Security Gateway will allow those connections through the cluster.

Find out whether or not the OPSEC certified clustering product can identify valid non-sticky connections.

- If the clustering product cannot identify valid non-sticky connections, the synchronization mechanism can do so instead. In that case, check Support non-sticky connections.
- If the clustering product can identify valid non-sticky connections, the synchronization mechanism does not have to take care of this. In that case, uncheck Support non-sticky connections. Usually it is safe to uncheck this option in High Availability solutions (not in Load Sharing). Clearing this option will lead to a slight improvement in the connection establishment rate.

If the Hide Cluster Members' outgoing traffic behind the Clusters IP Address option is checked, Support non-sticky connections should also be checked to support outgoing connections from a standby machine (unless specifically directed by OPSEC certified clustering product guide).

6. Many gateway clusters have a virtual cluster IP address that is defined in Topology page of the cluster object, in addition to physical cluster member interface addresses. The use of virtual cluster IP addresses affects the settings in the 3rd Party Configuration page.

When a client behind the cluster establishes an outgoing connection towards the Internet, the source address in the outgoing packets, is usually the physical IP address of the cluster member interface. If virtual cluster IP addresses are used, the clustering product usually changes the source IP address (using NAT) to that of the external virtual IP address of the cluster.

This corresponds to the default setting of Hide Cluster Members' outgoing traffic behind the Cluster's IP address being checked.

When a client establishes an incoming connection to the external virtual address of the cluster, the clustering product changes the destination IP address (using NAT) to that of the physical external address of one of the cluster members.

This corresponds to the default setting of Forward Cluster's incoming traffic to Cluster Members' IP addresses being checked. In the Topology page, define the interfaces for the individual members. In most OPSEC solutions, cluster IPs should not be added to the individual member's Topology tab. Refer to your clustering product documentation for additional information.

7. Define the other pages in the cluster object as required (NAT, VPN, Remote Access, and so on).


Note - When defining a Nokia cluster (VRRP or IP clustering) of IPSO version 3.9 and later, the monitor fw state feature should be disabled before the first policy installation. Failing to do so impedes the setting of the cluster IP addresses, and consequently the Get Interfaces operation in the Topology section of the Gateway Cluster Properties window will fail. After policy installation, the monitor fw state feature can be re-enabled.
CPHA Command Line Behavior in OPSEC Clusters

This section describes the behavior of specific command lines in OPSEC clusters.

Note - For details of the cpha command lines see Monitoring and Troubleshooting Gateway Clusters (on page 52).

The cphastart and cphastop Commands in OPSEC Clusters

The behavior of the cphastart and cphastop commands on ClusterXL clusters are described in The cphastart and cphastop Commands (on page 61).

On OPSEC clusters, the cphastart command may not cause the cluster member to start working. On IPSO clusters the behavior is the same as with ClusterXL clusters.

The cphastop command may not cause failover on OPSEC clusters. On IPSO IP Clustering clusters (but not on VRRP clusters), the behavior is the same as with ClusterXL clusters.

As with ClusterXL clusters, these commands should only be run by the Security Gateway, and not directly by the user.

The cphaprob Command in OPSEC Clusters

Use the cphaprob command to verify that the cluster and the cluster members are working properly. This command is relevant only for IPSO IP clustering and IPSO VRRP.

In non-IPSO OPSEC clusters the command output is either empty or the command does not have any effect.

To produce a usage printout for cphaprob that shows all the available commands, type cphaprob at the command line and press Enter. The meaning of each of these commands is explained in the following sections.

```
cphaprob -d <device> -t <timeout(sec)> -s <ok|init|problem> [-p] register
cphaprob -f <file> register
cphaprob -d <device> [-p] unregister
cphaprob -d <device> -s <ok|init|problem> report
```

```
cphaprob [-i[a]] [-e] list
```

```
cphaprob state
```

```
cphaprob [-a] if
```

**cphaprob state:** When running this command, the machine state is only Check Point status and is not really a machine status. The command only monitors full sync success, and if a policy was successfully installed. For IP clustering, the state is accurate and also includes the status of the IPSO Cluster. For VRRP, the status is accurate for a firewall, but it does not correctly reflect the status of the IPSO machine (for example, it does not detect interface failure).

```
cphaprob [-a] if: Shows only the relevant information - interface name, if it is a sync interface or not. "Multicast"/"Broadcast" refers to the cluster control protocol and is relevant only for the sync interface. Note that the status of the interface is not printed since it is not monitored. (This also applies in the IPSO machine.)
```
Chapter 7

UTM-1 Clustering

In This Chapter

Overview 46
Configuring a Cluster on New Appliances 46
Adding an Existing UTM-1 Appliance to a Cluster 49
Removing a Cluster Member 50
Upgrading to a UTM-1 Cluster 50
Importing a Database to a Primary Cluster Member 50
Migrating a Database to a UTM-1 Cluster 51
Supported Logging Options for UTM-1 Clusters 51

Overview

A pair of UTM-1 appliances can be clustered for high-availability. Each UTM-1 appliance becomes a single member in the cluster. High availability refers to the fact that both the gateway components and the Security Management server components are fully synchronized. If one component fails, a complete replication of the data exists on the member's peer in the cluster. Connections through the appliance continue uninterrupted.

Unlike between the gateway components, there is no fail-over between the Security Management server components. If the primary Security Management server goes down, the secondary Security Management server does not take over. However, the secondary Security Management server's database is fully synchronized with the Primary, so no data is lost. Before setting up a UTM-1 cluster, note that:

- The members of UTM-1 Cluster can either be configured together (both appliances are linked before the UTM-1 WebUI wizard is opened) or separately (the user chooses to figure a UTM-1 Cluster consisting of a single, primary member, and configure the secondary member at a later date).

- Even if you decide not to install a secondary cluster member, it is worthwhile to configure a cluster composed of a single primary member. A UTM-1 cluster is visible to the external network through its virtual IP addresses, not the actual physical addresses of its members. If at some point you do decide to add a secondary member, you will not have to alter the layer 3 topology of the network.

Configuring a Cluster on New Appliances

The following procedures describe how to configure a cluster on appliances with Messaging Security already installed.

To set up the appliance:
1. Power up the UTM-1 appliance.
   - The PWR/Status LED on the front panel starts blinking. When the LED ceases to blink, the appliance is ready for login.
2. Using the supplied Ethernet cable, connect the UTM-1 appliance's internal interface to a PC.
Configuring the IP Addresses

The PC and UTM-1 appliance must be on the same subnet.

To configure the PC to be on the same subnet as the UTM-1 appliance in Windows XP operating systems:
1. Click Start > Settings > Control Panel > Network Connections.
2. Double-click Local Area Connection.
4. In the Local Area Connection Properties window, scroll down to and double-click Internet Protocol (TCP/IP).
5. In the Internet Protocol (TCP/IP) Properties window that opens, select Use the following IP address.
6. In the IP address field, enter 192.168.1.2.
7. In the Subnet Mask field, enter 255.255.255.0.
8. Leave the Default gateway settings empty.
9. Select Use the following DNS server addresses, and leave the Preferred DNS server and Alternate DNS server fields empty. Click OK.

Initial Configuration

1. Launch your Web browser, and connect to https://192.168.1.2:4434. The UTM-1 Appliance login window appears.
   - If you receive a certificate error page notice, click Continue to this Web site.
   - If you are using a popup block, allow all popups.
2. Log in with the default system administrator login name and password: admin/admin.
3. Change the default login name and password
4. Click Save and Login.
5. Click Next. The Appliance Date and Time Setup page opens.
6. Set up the appliance date and time and click Apply. The Network Connections page opens.
7. Add an interface for the cluster SYNC/LAN1 interface, and click Next. The Routing Table page opens.
8. Add or delete routes as necessary, and click Next. The DNS and Domain Settings page opens.
9. Configure a host name, domain name, and DNS servers, and click Next. The Management Type page opens.
10. Select Locally Managed, and click Next.
11. The UTM-1 High Availability Cluster page opens.
12. Select UTM-1 Primary cluster member, and click Next.
   - If you are configuring the secondary Security Management Server, select UTM-1 Secondary Cluster member and click Next.
   The Secure Internal Communication (SIC) setup page opens,
   - Enter an activation key (one-time password), and make a note of it. You will need this activation key when initializing SIC in Security Management server. Click Next.
   The Web/SSH and GUI clients page opens.
13. Add or remove remote machines from which Web, SSH, or SmartConsole Clients can connect to the Security Management server, and click Next. The Download SmartConsole Applications page opens.
14. Download SmartConsole to your PC, and click Next. A summary page is displayed.
15. Click Finish. The configuration process runs.
16. Wait until the completed message appears, and click **OK**.
17. Reboot the gateway.
18. Repeat the procedure for the secondary Security Management Server, but at **step 14 on page 104**, select **UTM-1 Secondary cluster member**, and enter an activation key when prompted.

   **Note** - Remember the activation key. You will need it later when configuring the cluster in SmartDashboard. Remember to configure the cluster SYNC interface on the same subnet as the SYNC interface on the Primary Security Management Server.

You now have two UTM-1 appliances, one configured as a primary Security Management Server, the other as secondary.

**Configuring the Cluster in SmartDashboard**

1. Using a cross cable, connect the SYNC/LAN1 ports.
2. Install the SmartDashboard you downloaded in **step 16**.
3. Open SmartDashboard and connect to the IP address of the UTM-1 appliance that has been configured as the primary cluster member. The cluster configuration wizard starts automatically.
   
   If the UTM-1 Cluster wizard does not start automatically, double click the network cluster object in SmartDashboard and select the **Simple mode** option.
4. Click **Next** on the welcome page.
   
   The Cluster's general properties window opens:
5. Enter the cluster name.
   
   The name entered here will replace the provisional name that already appears in the network objects tree. (If you cancel the wizard at this point and examine the network objects tree, a cluster object is shown with only the primary member defined).

   **Note** - If you choose to define the UTM-1 Cluster in classic mode, when entering the IP address of the cluster, enter the Virtual IP.
6. Click **Next**.
   
   The **Secondary member's properties** window opens.
   
   The UTM-1 appliance acting as the primary Security Management server has been defined automatically, so only the secondary Security Management server needs to be configured.
7. Enter:
   
   a) The Secondary cluster's name and IP address, the same name that you entered during the SecurePlatform first-time configuration wizard.
   
   b) The activation key.
   
   If you choose to define the secondary member later, then each time SmartDashboard is opened, the UTM-1 first-time wizard runs automatically.
8. Click **Next**.
   
   Clicking **Next** initiates SIC. The Security Management server on the primary Security Management server retrieves a list of interfaces (and therefore networks) the secondary cluster member is connected to: the topology of the UTM-1 cluster.
   
   Once SIC is initiated, the **Cluster topology** window opens.
   
   Define member interfaces here, and also enter the Cluster interface IP address.
   
   - **Configure Cluster topology for interface** (SYNC, DMZ, EXT, INT, LANs)
     
     The LAN1 interface serves as the SYNC interface between cluster members. If not configured, SYNC interfaces are automatically set to 10.231.149.1 and 10.231.149.2. If these addresses are already in use, their values can be manually adjusted. If you manually adjust the default IP SYNC addresses, verify that both reside on the same subnet.
Adding an Existing UTM-1 Appliance to a Cluster

A standalone UTM-1 appliance can easily be added to a UTM-1 cluster after a second UTM-1 device is added to the network.

To add an existing UTM-1 device to a Cluster:
1. Open the SecurePlatform WebUI.
2. On the Product Configuration, Cluster page, select Make this Appliance the primary member of a UTM-1 High Availability Cluster.
3. Click Apply.
4. Reboot the appliance.
5. Using SmartDashboard, connect to the primary Security Management server.
   The first-time cluster configuration wizard opens.
6. Complete the wizard in order to configure the secondary cluster member.

In SmartDashboard, the network object representing the former standalone UTM-1 device is converted to a cluster object. If the standalone UTM-1 device appeared in the:

- INSTALL ON column of any rule (Security, QoS, NAT)
- Participating Gateways list for a VPN community

The new cluster object is now shown. In all other columns (Source, Destination, Groups) the standalone UTM-1 object needs to be manually changed to the cluster object.

Manually changing this network object may have implications for your Security, NAT, and QoS rules. To understand how and where the new primary cluster member is being used, right-click the cluster object and select Where used.

It is also recommended to use the Search > Query Rules option on the File Menu.
1. Select the Stand Alone object and make sure it appears in the In List section.
2. Select the Explicit option.
3. Click Apply.

For the remaining rules that still contain the standalone object, drag-and-drop the new cluster object.

Note - All interfaces in the cluster must have unique IP addresses. If the same IP address is used twice, policy installation will fail. A Load on gateway failed error message is displayed.

- Configure Cluster Virtual Interface.
  In addition to the UTM-1 cluster having its own unique IP address, which is visible to the network, each member interface also has a unique IP address. These unique IP addresses are used for internal communication between the cluster members as well as for accessing the cluster member directly. It is the Virtual IP address which makes the cluster visible to the external network, and populates the network routing tables — not the actual IP addresses of the individual members. If you chose to define a secondary member at a later date, but still configure the Virtual Cluster IP address, secondary members can later be added without changing the layer 3 topology of the network.

Note - The external Virtual IP address of the cluster must be unique.

9. Click Next, and in the following windows configure IP addresses for the External, and Internal interfaces on each cluster member.
10. Click Next, and Finish.

Note - After completing the first-time cluster wizard, the main IP address, as shown in the UTM-1 Cluster General properties window, will be the Virtual IP address of the cluster.

The new cluster of UTM-1 appliances appears in the network objects tree:
Removing a Cluster Member

Although a UTM-1 Cluster consists of only two members, it is still possible to remove one of the members without erasing the cluster. A UTM-1 cluster can exist with only a single (primary) member until a new secondary member is added. This means that either member can be easily taken offline for maintenance.

A cluster member is removed in two stages: first in the UTM-1 WebUI, second on the command line of the appliance. To remove a cluster member:

1. Open the UTM-1 WebUI > Product Configuration > Cluster page.
2. Click Remove Peer.
   Clicking Remove Peer results in a UTM-1 Cluster with a single member.
   • If the current machine is the primary member of the cluster, the secondary member is deleted.
   • If the current machine is the secondary member, the secondary member is first promoted to a primary and then the peer is deleted.
   
   **Note** - Services running on the appliance are restarted.
3. On the appliance command line, run: `cp_conf fullha disable`.
   This command reverts the primary cluster member to a standalone configuration.
4. Reboot.

After reverting to a standalone configuration, the former cluster is shown in SmartDashboard as a locally managed gateway consisting of gateway and Security Management server.

Upgrading to a UTM-1 Cluster

If you have a single UTM-1 appliance in a production environment, and wish to upgrade it to the current version and configure it as a UTM-1 Cluster:

1. Perform a standard in-place upgrade the current version using the UTM-1 WebUI.
2. Using the UTM-1 WebUI, on the Cluster page, convert the appliance to a primary member
3. Connect a second UTM-1 appliance to the network.
   1. If this second appliance is based on an earlier version, obtain the relevant upgrade package from the Download Center, save it to a USB stick, and reinstall the appliance as a secondary cluster member.
   2. If the second appliance is already R70, run the first-time wizard and select “secondary cluster member.”

Importing a Database to a Primary Cluster Member

To import an older Security Management server database to the primary member of a new UTM-1 cluster:

1. Install the primary as a standard Security Management server (not a cluster)
2. Import the older Security Management server database
3. Using the UTM-1 WebUI, convert the standard Security Management server to a primary cluster member.
Migrating a Database to a UTM-1 Cluster

To migrate a Security Management server database to a UTM-1 Cluster, go to the Check Point Support Center (http://support.checkpoint.com).

Supported Logging Options for UTM-1 Clusters

The standard logging options for a regular cluster are also available in UTM-1. In UTM-1, the primary cluster member can send logs to the secondary, and the secondary to the primary. Both can also send logs to an external log server. However, while a regular cluster cannot save logs locally, a UTM-1 Cluster member also functions as a log server. When logs are saved locally on the UTM-1 cluster member, implications exist for High Availability and Load Sharing scenarios.

Recommended Logging Options for High Availability

In a High Availability scenario, one of the UTM-1 cluster members is active while the other cluster member remains in standby. Log files are not synchronized between the two UTM-1 cluster members. For this reason it is recommended to:

- Configure logging so that logs are always sent to the primary UTM-1 cluster member, but to the secondary UTM-1 cluster member when the primary is unreachable.
- Set scheduled log forwarding to the primary UTM-1 cluster member.

Alternatively you can:

- Configure logging so that logs are sent to both UTM-1 cluster members. (SmartEvent and SmartReporter with standard reports should use only one of the cluster members as a source for log file correlation and consolidation). Or:
- Use an external log server.

Note - These logging options can also be applied to new gateways.

Load Sharing

To enable load sharing on a UTM-1 Cluster, you must first deselect the Save logs locally on each cluster member option on the Log Servers page.

When load sharing is enabled for a UTM-1 Cluster, a connection initiated to the primary member may terminate with the secondary. If logs were saved locally on each cluster member, only partial logs would be produced on each member for connections that were handled by both members. Saving logs locally would result in partial log data being displayed in Smartview Tracker.

If possible, when load sharing is enabled, configure log files to be sent to an external log server.

Note - If the older database is from another primary member of a UTM-1 Cluster, the above procedure is not required.
Chapter 8

Monitoring and Troubleshooting Gateway Clusters

In This Chapter

Verifying that a Cluster is Working Properly 52
Monitoring Cluster Status Using SmartConsole Clients 58
ClusterXL Configuration Commands 61
How to Initiate Failover 61
Monitoring Synchronization (fw ctl pstat) 62
Troubleshooting Synchronization 64
ClusterXL Error Messages 73
Member Fails to Start After Reboot 76

Verifying that a Cluster is Working Properly

The cphaprob Command

Use the cphaprob command to verify that the cluster and the cluster members are working properly, and to define critical devices. 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 failover 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:

- The cluster interfaces on the cluster members.
- Synchronization — full synchronization completed successfully.
- Filter — the Security Policy, and whether it is loaded.
- cphad — which follows the ClusterXL process called cphamcset.
- fwd — the Security Gateway daemon.

These commands can be run automatically by including them in scripts.

To produce a usage printout for cphaprob that shows all the available commands, type cphaprob at the command line and press Enter. The meaning of each of these commands is explained in the following sections.
cphaprob -d <device> -t <timeout(sec)> -s <ok|init|problem> [-p] register

cphaprob -f <file> register

cphaprob -d <device> [-p] unregister

cphaprob -d <device> [-s <ok|init|problem> report

cphaprob [-i[a]] [-e] list

cphaprob state

cphaprob [-a] if

### Monitoring Cluster Status

To see the status of a single or multiple cluster members:

- Run the following command:

  ```
  cphaprob state
  ```

  Do this command after setting up the cluster, and whenever you want to monitor the cluster status.

  The following is an example of the output of `cphaprob state`:

  ```
  cphaprob state

  Cluster mode: Load sharing (Multicast)

  Number Unique Address State
  1 (local) 30.0.0.1 active
  2 30.0.0.2 active
  ```

  - **Cluster mode** can be
    - Load Sharing (Multicast).
    - Load Sharing (Unicast).
    - High Availability New Mode (Primary Up or Active Up).
    - High Availability Legacy Mode (Primary Up or Active Up).
    - For third-party clustering products: “Service”, refer to Clustering Definitions and Terms (on page 10), for further information.
    - The number of the member indicates the member ID for Load Sharing, and the Priority for High Availability.
    - In Load sharing configuration, all machines in a fully functioning cluster should be Active. In High Availability configurations, only one machine in a properly functioning cluster must be Active, and the others must be in the Standby state.

  Third-party clustering products show Active/Active even if one of the members is in standby state. This is because this command only reports the status of the full synchronization process. For IPSO VRRP, this command shows the exact state of the Firewall, but not the cluster member (for example, the member may not be working properly but the state of the Firewall is active).

When examining the state of the cluster member, you need to consider whether it is forwarding packets, and whether it has a problem that is preventing it from forwarding packets. Each state reflects the result of a test on critical devices. This is a list that explains the possible cluster states, and whether or not they represent a problem.
### Table 8-9  **Cluster States**

<table>
<thead>
<tr>
<th>State</th>
<th>Meaning</th>
<th>Forwarding packets?</th>
<th>Is this state a problem?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>Everything is OK.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Active attention</td>
<td>A problem has been detected, but the cluster member is still forwarding packets because it is the only machine in the cluster or there is no other active machines in the cluster. In any other situation the state of the machine would be <em>down</em>.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Down</td>
<td>One of the critical devices is down.</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
| Ready                  | State *Ready* means that the machine recognizes itself as a part of the cluster and is literally ready to go into action, but, by design, something prevents the machine from taking action. Possible reasons that the machine is not yet *Active* include:  
1. Not all required software components were loaded and initialized yet and/or not all configuration steps finished successfully yet. Before a cluster member becomes *Active*, it sends a message to the rest of the cluster members, checking whether it can become *Active*. In High-Availability mode it will check if there is already an Active member and in Load Sharing Unicast mode it will check if there is a Pivot member already. The member remains in the *Ready* state until it receives the response from the rest of the cluster members and decides which state to choose next (*Active*, *Standby*, *Pivot*, or non-*Pivot*).  
2. Software installed on this member has a higher version than the rest of the members in this cluster. For example, when a cluster is upgraded from one version of Check Point Security Gateway to another, and the cluster members have different versions of Check Point Security Gateway, the members with a new version have the *Ready* state and the members with the previous version have the *Active / Active Attention* state.  
3. If the software installed on all cluster members includes CoreXL, which is installed by default in versions R70 and higher, a member in *Ready* state may have a higher number of CoreXL instances than other members. See sk42096 for a solution. | No                   | No                       |
| Standby                | Applies only to a High Availability configuration, and means the member is waiting for an active machine to fail in order to start packet forwarding. | No                   | No                       |
| Initializing           | An initial and transient state of the cluster member. The cluster member is booting up, and ClusterXL product is already running, but the Security Gateway is not yet ready. | No                   | No                       |
| ClusterXL inactive or machine is down | Local machine cannot hear anything coming from this cluster member. | Unknown              | Yes                      |
Monitoring Cluster Interfaces

To see the state of the cluster member interfaces and the virtual cluster interfaces:

- Run the following command on the cluster members:

  ```bash
  cphaprob [-a] if
  ```

  The output of this command must be identical to the configuration in the cluster object Topology page.

  For example:

  ```bash
  cphaprob -a if
  Required interfaces: 4
  Required secured interfaces: 1

  qfe4      UP       (secured, unique, multicast)
  qfe5      UP       (non secured, unique, multicast)
  qfe6      DOWN (4810.2 secs) (non secured, unique, multicast)
  qfe7      UP       (non secured, unique, multicast)

  Virtual cluster interfaces: 2
  qfe5      30.0.1.130
  qfe6      30.0.2.130
  ```

  The interfaces are ClusterXL critical devices. ClusterXL checks the number of good interfaces and sets a value of Required interfaces to the maximum number of good interfaces seen since the last reboot. If the number of good interfaces is less than the Required number, ClusterXL initiates failover. The same for secured interfaces, where only the good synchronization interfaces are counted.

  An interface can be:

  - **Non-secured** or **Secured**. A secured interface is a synchronization interface.
  - **Shared** or **unique**. A shared interface applies only to High Availability Legacy mode.
  - **Multicast** or **broadcast**. The Cluster Control Protocol (CCP) mode used in the cluster. CCP can be changed to use broadcast instead. To toggle between these two modes use the command `cphaconf set_ccp <broadcast|multicast>`

  For third-party clustering products, except in the case of IPSO IP Clustering, `cphaprob -a if` should always show virtual cluster IP addresses.

  When an interface is DOWN, it means that the interface cannot receive or transmit CCP packets, or both. This may happen when an interface is malfunctioning, is connected to an incorrect subnet, is unable to pick up Multicast Ethernet packets and so on. The interface may also be able to receive but not transmit CCP packets, in which case the status field is read. The displayed time is the number of seconds that have elapsed since the interface was last able to receive/transmit a CCP packet.

  See Defining Disconnected Interfaces (on page 96) for additional information.

Monitoring Critical Devices

When a critical device fails, the cluster member is considered to have failed. To see the list of critical devices on a cluster member, and of all the other machines in the cluster, run the following command on the cluster member:
There are a number of built-in critical devices, and the administrator can define additional critical devices. The default critical devices are:

- The cluster interfaces on the cluster members.
- **Synchronization** — full synchronization completed successfully.
- **Filter** — the Security Policy, and whether it is loaded.
- **cphad** — which follows the ClusterXL process called **cphamcset**.
- **fwd** — the Security Gateway daemon.

For IPSO Clustering, the output is the same as for ClusterXL Load Sharing. For other third-party products, this command produces no output. The following example output shows that the **fwd** process is down:

```
cphaprob list

Built-in Devices:

Device Name: Interface Active Check
Current state: OK

Registered Devices:

Device Name: Synchronization
Registration number: 0
Timeout: none
Current state: OK
Time since last report: 15998.4 sec

Device Name: Filter
Registration number: 1
Timeout: none
Current state: OK
Time since last report: 15644.4 sec

Device Name: fwd
Registration number: 3
Timeout: 2 sec
Current state: problem
Time since last report: 4.5 sec
```

**Registering a Critical Device**

```
cphaprob [-i[a]] [-e] list

cphaprob [-i[a]] [-e] [-p] register
```
It is possible to add a user defined critical device to the default list of critical devices. Use this command to 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. If <device> fails, then the cluster member is considered to have failed. If <device> fails to contact the cluster member in <timeout> seconds, <device> will be considered to have failed. For no timeout, use the value 0.

Define the status of the <device> that will be reported to ClusterXL upon registration. This initial status can be one of:

- ok — <device> is alive.
- init — <device> is initializing. The machine is down. This state prevents the machine from becoming active.
- problem — <device> has failed.

[-p] makes these changes permanent. After performing a reboot or after removing the Security Gateway (on Linux or IPSO for example) and re-attaching it, the status of critical devices that were registered with this flag will be saved.

### Registering Critical Devices Listed in a File

**cphaprob -f <file> register**

Register all the user defined critical devices listed in <file>. <file> must be an ASCII file, with each device on a separate line. Each line must list three parameters, which must be separated by at least a space or a tab, as follows:

<device> <timeout> <status>

- <device> — The name of the critical device. It must have no more than 15 characters, and must not include white spaces.
- <timeout> — If <device> fails to contact the cluster member in <timeout> seconds, <device> will be considered to have failed. For no timeout, use the value 0.
- <status> — can be one of
  - ok — <device> is alive.
  - init — <device> is initializing. The machine is down. This state prevents the machine from becoming active.
  - problem — <device> has failed.

### Unregistering a Critical Device

**cphaprob -d <device> [-p] unregister**

Unregistering a user defined <device> as a critical process. This means that this device is no longer considered critical. If a critical device (and hence a cluster member) was registered as "problem" before running this command, then after running this command the status of the cluster will depend only on the remaining critical devices.

[-p] makes these changes permanent. This means that after performing a reboot or after removing the kernel (on Linux or IPSO for example) and re-attaching it, these critical devices remain unregistered.

### Reporting Critical Device Status to ClusterXL

**cphaprob -d <device> -s <ok|init|problem> report**

Use this command to report the status of a user defined critical device to ClusterXL.
<device> is the device that must be running for the cluster member to be considered active. If <device> fails, then the cluster member is considered to have failed.

The status to be reported. The status can be one of:

- **ok** — <device> is alive
- **init** — <device> is initializing. The machine is down. This state prevents the machine from becoming active.
- **problem** — <device> has failed. If this status is reported to ClusterXL, the cluster member will immediately failover to another cluster member.

If <device> fails to contact the cluster member within the timeout that was defined when it was registered, <device> and hence the cluster member, will be considered to have failed. This is true only for critical devices with timeouts. If a critical device is registered with the `-t 0` parameter, there will be no timeout, and until the device reports otherwise, the status is considered to be the last reported status.

### Example cphaprob Script

Predefined **cphaprob** scripts are located on the location `$FWDIR/bin`. Two scripts are available:

- `clusterXL_monitor_ips`
- `clusterXL_monitor_process`

The **clusterXL_monitor_ips** script in the Appendix chapter Example cphaprob Script (on page 116) has been designed to provide a way to check end-to-end connectivity to routers or other network devices and cause failover if the ping fails. The **clusterXL_monitor_process** script monitors the existence of given processes and causes failover if the processes die. This script uses the normal `pnote` mechanism.

### Monitoring Cluster Status Using SmartConsole Clients

#### SmartView Monitor

SmartView Monitor displays a snapshot of all ClusterXL cluster members in the enterprise, enabling real-time monitoring and alerting. For each cluster member, state change and critical device problem notifications are displayed. SmartView Monitor allows you to specify the action to be taken if the status of a cluster member changes. For example, the Security Gateway can issue an alert notifying you of suspicious activity.

#### Starting and Stopping ClusterXL Using SmartView Monitor

To stop ClusterXL on the machine and cause failover to another machine, open SmartView Monitor, click the cluster object, select one of the member gateway branches, right click a cluster member, and select **Down**.

To initiate a restart of ClusterXL, open SmartView Monitor, click the cluster object, select one of the member gateway branches, right click a cluster member, and select **Up**.

**Note** - SmartView Monitor does not initiate full synchronization, so that some connections may be lost. To initiate full synchronization, run `cpstart`.

#### SmartView Tracker

Every change in status of a cluster member is recorded in SmartView Tracker according to the choice in the **Fail-Over Tracking** option of the cluster object **ClusterXL** page.
**ClusterXL Log Messages**

The following conventions are used in this section:

1. Square brackets are used to indicate place holders, which are substituted by relevant data when an actual log message is issued (for example, [NUMBER] will be replaced by a numeric value).
2. Angle brackets are used to indicate alternatives, one of which will be used in actual log messages. The different alternatives are separated with a vertical line (for example, <up|down> indicates that either "up" or "down" will be used).
3. The following place holders are frequently used:
   - ID: A unique cluster member identifier, starting from "1". This corresponds to the order in which members are sorted in the cluster object's GUI.
   - IP: Any unique IP address that belongs to the member.
   - MODE: The cluster mode (for example, New HA, LS Multicast, and so on).
   - STATE: The state of the member (for example, active, down, standby).
   - DEVICE: The name of a pnote device (for example, fwd, Interface Active Check).

**General logs**

Starting <ClusterXL|State Synchronization>.

Indicates that ClusterXL (or State Synchronization, for 3rd party clusters) was successfully started on the reporting member. This message is usually issued after a member boots, or after an explicit call to cphastart.

Stopping <ClusterXL|State Synchronization>.

Informs that ClusterXL (or State Synchronization) was deactivated on this machine. The machine will no longer be a part of the cluster (even if configured to be so), until ClusterXL is restarted.

Unconfigured cluster Machines changed their MAC Addresses. Please reboot the cluster so that the changes take affect.

This message is usually issued when a machine is shut down, or after an explicit call to cphastop.

**State logs**

Mode inconsistency detected: member [ID] ([IP]) will change its mode to [MODE]. Please re-install the security policy on the cluster.

This message should rarely happen. It indicates that another cluster member has reported a different cluster mode than is known to the local member. This is usually the result of a failure to install the security policy on all cluster members. To correct this problem, install the Security Policy again.

*Note* - The cluster will continue to operate after a mode inconsistency has been detected, by altering the mode of the reporting machine to match the other cluster members. However, it is highly recommended that the policy will be re-installed as soon as possible.

State change of member [ID] ([IP]) from [STATE] to [STATE] was cancelled, since all other members are down. Member remains [STATE].

When a member needs to change its state (for example, when an active member encounters a problem and needs to bring itself down), it first queries the other members for their state. If all other members are down, this member cannot change its state to a non-active one (or else all members will be down, and the cluster will not function). Thus, the reporting member continues to function, despite its problem (and will usually report its state as "active attention").

member [ID] ([IP]) <is active|is down|is stand-by|is initializing> ([REASON]).

This message is usually issued whenever a cluster member changes its state. The log text specifies the new state of the member.

**Pnote logs**

PNote log messages are issued whenever a pnote device changes its state.
- **[DEVICE] on member [ID] ([IP]) status OK ([REASON]).**
  The pnote device is working normally.

- **[DEVICE] on member [ID] ([IP]) detected a problem ([REASON]).**
  Either an error was detected by the pnote device, or the device has not reported its state for a number of seconds (as set by the "timeout" option of the pnote).

- **[DEVICE] on member [ID] ([IP]) is initializing ([REASON]).**
  Indicates that the device has registered itself with the pnote mechanism, but has not yet determined its state.

- **[DEVICE] on member [ID] ([IP]) is in an unknown state ([STATE ID]) ([REASON]).**
  This message should not normally appear. Contact Check Point Support.

**Interface logs**

- **interface [INTERFACE NAME] of member [ID] ([IP]) is up.**
  Indicates that this interface is working normally, meaning that it is able to receive and transmit packets on the expected subnet.

- **interface [INTERFACE NAME] of member [ID] ([IP]) is down (receive <up|down>, transmit <up|down>).**
  This message is issued whenever an interface encounters a problem, either in receiving or transmitting packets. Note that in this case the interface may still be working properly, as far as the OS is concerned, but is unable to communicate with other cluster members due to a faulty cluster configuration.

- **interface [INTERFACE NAME] of member [ID] ([IP]) was added.**
  Notifies users that a new interface was registered with the Security Gateway (meaning that packets arriving on this interface are filtered by the firewall). Usually this message is the result of activating an interface (such as issuing an `ifconfig up` command on Unix systems). The interface will now be included in the ClusterXL reports (such as in SmartView Monitor, or in the output of `cphaprob -a if`). Note that the interface may still be reported as "Disconnected", in case it was configured as such for ClusterXL.

- **interface [INTERFACE NAME] of member [ID] ([IP]) was removed.**
  Indicates that an interface was detached from the Security Gateway, and is therefore no longer monitored by ClusterXL.

**SecureXL logs**

- **SecureXL device was deactivated since it does not support CPLS.**
  This message is the result of an attempt to configure a ClusterXL in Load Sharing Multicast mode over Security Gateways using an acceleration device that does not support Load Sharing. As a result, acceleration will be turned off, but the cluster will work in Check Point Load Sharing mode (CPLS).

**Reason Strings**

- **member [ID] ([IP]) reports more interfaces up.**
  This text can be included in a pnote log message describing the reasons for a problem report: Another member has more interfaces reported to be working, than the local member does. This means that the local member has a faulty interface, and that its counterpart can do a better job as a cluster member. The local member will therefore go down, leaving the member specified in the message to handle traffic.

- **member [ID] ([IP]) has more interfaces - check your disconnected interfaces configuration in the <discntd.if file|registry>.**
  This message is issued when members in the same cluster have a different number of interfaces. A member having less interfaces than the maximal number in the cluster (the reporting member) may not be working properly, as it is missing an interface required to operate against a cluster IP address, or a synchronization network. If some of the interfaces on the other cluster member are redundant, and should not be monitored by ClusterXL, they should be explicitly designated as "Disconnected". This is done using the file `$FWDIR/conf/discntd.if` (under Unix systems), or the Windows Registry.

- **[NUMBER] interfaces required, only [NUMBER] up.**
  ClusterXL has detected a problem with one or more of the monitored interfaces. This does not necessarily mean that the member will go down, as the other members may have less operational
interfaces. In such a condition, the member with the highest number of operational interfaces will remain up, while the others will go down.

**ClusterXL Configuration Commands**

**The cphaconf Command**

Running this command is not recommended. It should be run only by the Security Gateway.

```
cphaconf [-i <machine id>] [-p <policy id>] [-b <db_id>] [-n <cluster num>] [-c <cluster size>] [-m <service >]
            [-t <secured IF 1>...] start

cphaconf [-t <secured IF 1>...] [-d <disconnected IF 1>...] add

cphaconf clear-secured

cphaconf clear-disconnected

cphaconf stop

cphaconf init

cphaconf forward <on/off>

cphaconf debug <on/off>

cphaconf set_ccp <broadcast/multicast>

cphaconf mc_reload

cphaconf debug_data
```

**The cphastart and cphastop Commands**

Running **cphastart** on a cluster member activates ClusterXL on the member. It does not initiate full synchronization. **cpstart** is the recommended way to start a cluster member.

Running **cphastop** on a cluster member stops the cluster member from passing traffic. State synchronization also stops. It is still possible to open connections directly to the cluster member. In High Availability Legacy mode, running **cphastop** may cause the entire cluster to stop functioning.

These commands should only be run by the Security Gateway, and not directly by the user.

**How to Initiate Failover**

The state of a cluster member can be manually controlled in order to take down the cluster member. This initiates failover to the other cluster member(s), in the case of Load Sharing, or failover to the next highest priority cluster member in the case of High Availability.

**Stopping the Cluster Member**

To stop ClusterXL on the machine and cause failover to another machine:
1. Register a dummy critical device (**faildevice** for example) using the command `cphaprobc -d faildevice -t 0 -s ok register`.
2. Then run the following command to report to ClusterXL that the critical device **faildevice** has a problem: `cphaprobc -d faildevice -s problem report`. Failover to another cluster member will immediately occur.

Alternatively:
1. Open SmartView Monitor, and click the cluster object.
2. Select one of the member gateway branches.
3. Right click a cluster member, and then select **Down**.

**Starting the Cluster Member**

ClusterXL starts automatically when the Security Gateway is started on the cluster member (**cpstart**).

**To initiate a restart of ClusterXL:**

- If the cluster member was downed using the command `cphaprob -d faildevice -s problem report`, run either of the following commands:
  - `cphaprob -d faildevice -s ok report`
  - `cphaprob -d faildevice unregister`

Alternatively,

1. Open SmartView Monitor, and click the cluster object.
2. Select the member gateway branch that appears as "Down".
3. Right click on that cluster member, and select **Up**.

**Note** - Starting the Cluster member from SmartView Monitor does not initiate full synchronization, so some connections may be lost. To initiate full synchronization, perform **cpstart**.

**Monitoring Synchronization (fw ctl pstat)**

**To monitor the synchronization mechanism on ClusterXL or third-party OPSEC certified clustering products:**

- Run the following command on a cluster member

```plaintext
fw ctl pstat
```

The output of this command is a long list of statistics for the Security Gateway. At the end of the list there is a section called "Synchronization" which applies per Gateway Cluster member. Many of the statistics are counters that can only increase. A typical output is as follows:

```
Version: new
Status: Able to Send/Receive sync packets
Sync packets sent:
    total : 3976, retransmitted : 0, retrans reqs : 58, acks : 97
Sync packets received:
    total : 4290, were queued : 58, dropped by net : 47
    retrans reqs : 0, received 0 acks
    retrans reqs for illegal seq : 0
Callback statistics: handled 3 cb, average delay : 1, max delay : 2
Delta Sync memory usage: currently using XX KB mem
Callback statistics: handled 322 cb, average delay : 2, max delay : 8
Number of Pending packets currently held: 1
Packets released due to timeout: 18
```
The meaning of each line in this printout is explained below.

**Version: new**

This line must appear if synchronization is configured. It indicates that new sync is working (as opposed to old sync from version 4.1).

**Status: Able to Send/Receive sync packets**

If sync is unable to either send or receive packets, there is a problem. Sync may be temporarily unable to send or receive packets during boot, but this should not happen during normal operation. When performing full sync, sync packet reception may be interrupted.

**Sync packets sent:**

```
  total : 3976,  retransmitted : 0, retrans reqs : 58,  acks : 97
```

The total number of sync packets sent is shown. Note that the total number of sync packets is non-zero and increasing.

The cluster member sends a retransmission request when a sync packet is received out of order. This number may increase when under load.

**Acks** are the acknowledgements sent for received sync packets, when an acknowledgement was requested by another cluster member.

**Sync packets received:**

```
  total : 4290,  were queued : 58, dropped by net : 47
```

The total number of sync packets received is shown. The queued packets figure increases when a sync packet is received that complies with one of the following conditions:

1. The sync packet is received with a sequence number that does not follow the previously processed sync packet.
2. The sync packet is fragmented. This is done to solve MTU restrictions.

This figure never decreases. A non-zero value does not indicate a problem.

The **dropped by net** number may indicate network congestion. This number may increase slowly under load. If this number increases too fast, a networking error may be interfering with the sync protocol. In that case, check the network.

```
  retrans reqs : 0, received 0 acks
  retrans reqs for illegal seq : 0
  Callback statistics: handled 3 cb, average delay : 1,  max delay : 2
```

This message refers to the number of received retransmission requests, in contrast to the transmitted retransmission requests in the section above. When this number grows very fast, it may indicate that the load on the machine is becoming too high for sync to handle.

**Acks** refer to the number of acknowledgements received for the "cb request" sync packets, which are sync packets with requests for acknowledgments.

**Retrans reqs for illegal seq** displays the number of retransmission requests for packets which are no longer in this member's possession. This may indicate a sync problem.

**Callback statistics** relate to received packets that involve Flush and Ack. This statistic only appears for a non-zero value.
The callback **average delay** is how much the packet was delayed in this member until it was released when the member received an ACK from all the other members. The delay happens because packets are held until all other cluster members have acknowledged reception of that sync packet.

This figure is measured in terms of numbers of packets. Normally this number should be small (~1-5). Larger numbers may indicate an overload of sync traffic, which causes connections that require sync acknowledgements to suffer slight latency.

```plaintext
dropped updates as a result of sync overload: 0
```

In a heavily loaded system, the cluster member may drop synchronization updates sent from another cluster member.

```plaintext
Delta Sync memory usage: currently using XX KB mem
```

**Delta Sync memory usage** only appears for a non-zero value. Delta sync requires memory only while full sync is occurring. Full sync happens when the system goes up - after reboot for example. At other times, Delta sync requires no memory because Delta sync updates are applied immediately. For information about Delta sync see How State Synchronization Works (on page 12).

```plaintext
Number of Pending packets currently held: 1
Packets released due to timeout: 18
```

**Number of Pending packets currently held** only appears for a non-zero value. ClusterXL prevents out-of-state packets in non-sticky connections. It does this by holding packets until a SYN-ACK is received from all other active cluster members. If for some reason a SYN-ACK is not received, the Security Gateway on the cluster member will not release the packet, and the connection will not be established.

**Packets released due to timeout** only appears for a non-zero value. If the Number of Pending Packets is large (more than 100 pending packets), and the number of Packets released due to timeout is small, you should take action to reduce the number of pending packets. To solve this problem, see Reducing the Number of Pending Packets (on page 95).

---

**Troubleshooting Synchronization**

**Introduction to cphaprob [-reset] syncstat**

Heavily loaded clusters and clusters with geographically separated members pose special challenges. High connection rates, and large distances between the members can lead to delays that affect the operation of the cluster.

The **cphaprob [-reset] syncstat** command is a tool for monitoring the operation of the State Synchronization mechanism in highly loaded and distributed clusters. It can be used for both ClusterXL and third-party OPSEC certified clustering products.

The troubleshooting process is as follows:

1. Run the `cphaprob syncstat` command.
2. Examine and understand the output statistics.
3. Tune the relevant synchronization global configuration parameters.
4. Rerun the command, **resetting** the statistics counters using the -reset option:
   ```plaintext
cphaprob -reset syncstat
```
5. Examine the output statistics to see if the problem is solved.

The section Output of cphaprob [-reset] syncstat (on page 65) explains each of the output parameters, and also explains when the output represents a problem.

Any identified problem can be solved by performing one or more of the tips described in **Synchronization Troubleshooting Options** (on page 71).
Output of cphaprob [-reset] syncstat

The output parameters of the `cphaprob syncstat` command are shown below. The values (not shown) give an insight into the state and characteristics of the synchronization network. Each parameter and the meaning of its possible values is explained in the following sections.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sync Statistics (IDs of F&amp;A Peers - 1):</td>
<td>147</td>
</tr>
<tr>
<td>Other Member Updates:</td>
<td>147</td>
</tr>
<tr>
<td>Sent Retransmission Requests</td>
<td>147</td>
</tr>
<tr>
<td>Avg</td>
<td>Missing Updates per Request</td>
</tr>
<tr>
<td>Old or too-new Arriving Updates</td>
<td>148</td>
</tr>
<tr>
<td>Unsynchronized Missing Updates</td>
<td>148</td>
</tr>
<tr>
<td>Lost Sync Connection (num of events)</td>
<td>149</td>
</tr>
<tr>
<td>Timed out Sync Connection</td>
<td>149</td>
</tr>
<tr>
<td>Local Updates</td>
<td>149</td>
</tr>
<tr>
<td>Total Generated Updates</td>
<td>150</td>
</tr>
<tr>
<td>Recv Retransmission requests</td>
<td>150</td>
</tr>
<tr>
<td>Recv Duplicate Retrans request</td>
<td>150</td>
</tr>
<tr>
<td>Blocking Scenarios</td>
<td>151</td>
</tr>
<tr>
<td>Blocked Packets</td>
<td>152</td>
</tr>
<tr>
<td>Max Length of Sending Queue</td>
<td>152</td>
</tr>
<tr>
<td>Avg Length of Sending Queue</td>
<td>153</td>
</tr>
<tr>
<td>Hold Pkts Events</td>
<td>154</td>
</tr>
<tr>
<td>Unhold Pkt Events</td>
<td>154</td>
</tr>
<tr>
<td>Not Held Due to no Members</td>
<td>154</td>
</tr>
<tr>
<td>Max Held Duration (ticks)</td>
<td>155</td>
</tr>
<tr>
<td>Avg Held Duration (ticks)</td>
<td>155</td>
</tr>
<tr>
<td>Timers:</td>
<td>156</td>
</tr>
<tr>
<td>Sync tick (ms)</td>
<td>156</td>
</tr>
<tr>
<td>CPHA tick (ms)</td>
<td>156</td>
</tr>
<tr>
<td>Queues:</td>
<td>156</td>
</tr>
<tr>
<td>Sending Queue Size</td>
<td>156</td>
</tr>
<tr>
<td>Receiving Queue Size</td>
<td>156</td>
</tr>
</tbody>
</table>

**Sync Statistics (IDs of F&A Peers - 1):**

These statistics relate to the state synchronization mechanism. The F&A (Flush and Ack) peers are the cluster members that this member recognizes as being part of the cluster. The IDs correspond to IDs and IP addresses generated by the `cphaprob state` command.
**Other Member Updates:**

The statistics in this section relate to *updates* generated by other cluster members, or to updates that were not received from the other members. Updates inform about changes in the connections handled by the cluster member, and are sent from and to members. Updates are identified by sequence numbers.

**Sent Retransmission Requests**

The number of retransmission requests, which were sent by this member. Retransmission requests are sent when certain packets (with a specified sequence number) are missing, while the sending member already received updates with advanced sequences.

A high value can imply connectivity problems.

- **Note** - Compare the number of retransmission requests to the Total Regenerated Updates of the other members (see Total Generated Updates on page 67).

  If its value is unreasonably high (more than 30% of the Total Generated Updates of other members), contact Technical Support equipped with the entire output and a detailed description of the network topology and configuration.

**Avg | Missing Updates per Request**

Each retransmission request can contain up to 32 missing consecutive sequences. The value of this field is the average number of requested sequences per retransmission request.

More than 20 missing consecutive sequences per retransmission request can imply connectivity problems.

- **Note** - If this value is unreasonably high, contact Technical Support, equipped with the entire output and a detailed description of the network topology and configuration.

**Old or too-new Arriving Updates**

The number of arriving sync updates where the sequence number is too low, which implies it belongs to an old transmission, or too high, to the extent that it cannot belong to a new transmission.

Large values imply connectivity problems.

- **Note** - See Enlarging the Receiving Queue (on page 72) If this value is unreasonably high (more than 10% of the total updates sent), contact Technical Support, equipped with the entire output and a detailed description of the network topology and configuration.

**Unsynced Missing Updates**

The number of missing sync updates for which the receiving member stopped waiting. It stops waiting when the difference in sequence numbers between the newly arriving updates and the missing updates is larger than the length of the receiving queue.

This value should be zero. However, the loss of some updates is acceptable as long as the number of lost updates is less than 1% of the total generated updates.
**Troubleshooting Synchronization**

**Monit and Troubleshooting Gateway Clusters**

Page 67

**Note** - To decrease the number of lost updates, expand the capacity of the Receiving Queue. See Enlarging the Receiving Queue (on page 72).

**Lost Sync Connection (num of events)**

The number of events in which synchronization with another member was lost and regained due to either Security Policy installation on the other member, or a large difference between the expected and received sequence number.

The value should be zero. A positive value indicates connectivity problems.

**Note** - Allow the sync mechanism to handle large differences in sequence numbers by expanding the Receiving Queue capacity. See Enlarging the Receiving Queue (on page 72).

**Timed out Sync Connection**

The number of events in which the member declares another member as not connected. The member is considered as disconnected because no ACK packets were received from that member for a period of time (one second), even though there are Flush and Ack packets being held for that member.

The value should be zero. Even with a round trip time on the sync network as high as 100ms, one second should be enough time to receive an ACK. A positive value indicates connectivity problems.

**Note** - Try enlarging the Sync Timer (see Enlarging the Sync Timer (on page 72)). However, you may well have to contact Technical Support equipped with the entire output and a detailed description of the network topology and configuration.

**Local Updates**

The statistics in this section relate to updates generated by the local cluster member. Updates inform about changes in the connections handled by the cluster member, and are sent from and to members. Updates are identified by sequence numbers.

**Total Generated Updates**

The number of sync update packets generated by the sync mechanism since the statistics were last reset. Its value is the same as the difference between the sequence number when applying the -reset option, and the current sequence number.

Can have any value.

**Recv Retransmission requests**

The number of received retransmission requests. A member requests retransmissions when it is missing specified packets with lower sequence numbers than the ones already received.

A large value can imply connectivity problems.

**Note** - If this value is unreasonably high (more than 30% of the Total Generated Updates) contact Technical Support, equipped with the entire output and a detailed description of the network topology and configuration.
**Recv Duplicate Retrans request**

The number of duplicated retransmission requests received by the member. Duplicate requests were already handled, and so are dropped.

A large value may indicate network problem or storms on the sync network.

*Note* - If this value is unreasonably high (more than 30% of the Total Generated Updates) contact Technical Support, equipped with the entire output and a detailed description of the network topology and configuration.

---

**Blocking Scenarios**

Under extremely heavy load conditions, the cluster may block new connections. This parameter shows the number of times that the cluster member started blocking new connections due to sync overload.

The member starts to block connections when its Sending Queue has reached its capacity threshold. The capacity threshold is calculated as 80% of the difference between the current sequence number and the sequence number for which the member received an ACK from all the other operating members.

A positive value indicates heavy load. In this case, observe the *Blocked Packets* to see how many packets we blocked. Each dropped packet means one blocked connection.

This parameter is only measured if the *Block New Connections* mechanism (described in Blocking New Connections Under Load (on page 94)) is active.

**To activate the Block New Connections mechanism:**

- Apply the `fw ctl set int fw_sync_block_new_conns 0` command to all the cluster members.

  *Note* - The best way to handle a severe blocking connections problem is to enlarge the sending queue. See Enlarging the Sending Queue (on page 71).

  Another possibility is to decrease the timeout after which a member initiates an ACK. See Reconfiguring the Acknowledgment Timeout (on page 72). This updates the sending queue capacity more accurately, thus making the blocking process more precise.

---

**Blocked Packets**

The number of packets that were blocked because the cluster member was blocking all new connections (see *Blocking Scenarios*). The number of blocked packets is usually one packet per new connection attempt.

A value higher than 5% of the Sending Queue (see Avg Length of Sending Queue (on page 69)) can imply a connectivity problem, or that ACKs are not being sent frequently enough.

This parameter is only measured if the *Block New Connections* mechanism (described in Blocking New Connections Under Load (on page 94)) is active.

**To activate the Block New Connections mechanism:**

- Apply the `fw ctl set int fw_sync_block_new_conns 0` command on all the cluster members.
Note - The best way to handle a severe blocking connections problem is to enlarge the sending queue. See Enlarging the Sending Queue (see "Enlarging the Sync Timer" on page 72).

Another possibility is to decrease the timeout after which a member initiates an ACK. See Reconfiguring the Acknowledgment Timeout (on page 72). This updates the sending queue capacity more accurately, thus making the blocking process more precise.

**Max Length of Sending Queue**

The size of the Sending Queue is fixed. By default it is 512 sync updates. As newer updates with higher sequence numbers enter the queue, older updates with lower sequence numbers drop off the end of the queue. An older update could be dropped from the queue before the member receives an ACK about that update from all the other members.

This parameter is the difference between the current sync sequence number and the last sequence number for which the member received an ACK from all the other members. The value of this parameter can therefore be greater than 512.

The value of this parameter should be less than 512. If larger than 512, there is not necessarily a sync problem. However, the member will be unable to answer retransmission request for updates which are no longer in its queue.

This parameter is only measured if the *Block New Connections* mechanism (described in Blocking New Connections Under Load (on page 94)) is active.

**To activate the Block New Connections mechanism:**

- Apply the `fw ctl set int fw_sync_block_new_conns 0` command on all the cluster members.

Note - Enlarge the Sending Queue to value larger than this value. See Enlarging the Sending Queue (on page 71).

**Avg Length of Sending Queue**

The average value of the *Max Length of Sending Queue* parameter, since reboot or since the Sync statistics were reset.

The value should be up to 80% of the size of the Sending Queue.

This parameters is only measured if the *Block New Connections* mechanism (described in Blocking New Connections Under Load (on page 94)) is active.

**To activate the Block New Connections mechanism:**

- Apply the `fw ctl set int fw_sync_block_new_conns 0` command on all the cluster members.

Note - Enlarge the Sending Queue so that this value is not larger than 80% of the new queue size. See Enlarging the Sending Queue (on page 71).

**Hold Pkts Events**

The number of occasions where the sync update required Flush and Ack, and so was kept within the system until an ACK arrived from all the other functioning members.

Should be the same as the number of *Unhold Pkt Events*.
Troubleshooting Synchronization

Monitoring and Troubleshooting Gateway Clusters

Page 70

Note - Contact Technical Support equipped with the entire output and a detailed description of the network topology and configuration.

Unhold Pkt Events

The number of occasions when the member received all the required ACKS from the other functioning members.

Should be the same as the number of Hold Pkts Events.

Note - Contact Technical Support equipped with the entire output and a detailed description of the network topology and configuration.

Not Held Due to no Members

The number of packets which should have been held within the system, but were released because there were no other operating members.

When the cluster has at least two live members, the value should be 0.

Note - The cluster has a connectivity problem. Examine the values of the parameters: Lost Sync Connection (num of events) (on page 67) and Timed out Sync Connection (on page 67) to find out why the member thinks that it is the only cluster member.

You may also need to contact Technical Support equipped with the entire output and a detailed description of the network topology and configuration.

Max Held Duration (ticks)

The maximum time in ticks (one tick equals 100ms) for which a held packet was delayed in the system for Flush and Ack purposes.

It should not be higher than 50 (5 seconds), because of the pending timeout mechanism which releases held packets after a certain timeout. By default, the release timeout is 50 ticks. A high value indicates connectivity problem between the members.

Note - Optionally change the default timeout by changing the value of the fwlidbcast_pending_timeout global variable. See Advanced Cluster Configuration (on page 93) and Reducing the Number of Pending Packets (on page 95).

Also, examine the parameter Timed out Sync Connection (on page 67) to understand why packets were held for a long time.

You may also need to contact Technical Support equipped with the entire output and a detailed description of the network topology and configuration.

Avg Held Duration (ticks)

The average duration in ticks (tick equals 100ms) that held packets were delayed within the system for Flush and Ack purposes.

The average duration should be about the round-trip time of the sync network. A larger value indicates connectivity problem.
**Note** - If the value is high, contact Technical Support equipped with the entire output and a detailed description of the network topology and configuration in order to examine the cause to the problem.

**Timers:**

The Sync and CPHA timers perform sync and cluster related actions every fixed interval.

**Sync tick (ms)**

The *Sync timer* performs cluster related actions every fixed interval. By default, the Sync timer interval is 100ms. The base time unit is 100ms (or 1 tick), which is also the minimum value.

**CPHA tick (ms)**

The *CPHA timer* performs cluster related actions every fixed interval. By default, the CPHA timer interval is 100ms. The base time unit is 100ms (or 1 tick), which is also the minimum value.

**Queues:**

Each cluster member has two queues. The Sending Queue and the Receiving Queue.

**Sending Queue Size**

The Sending Queue on the cluster member stores locally generated sync updates. Updates in the Sending Queue are replaced by more recent updates. In a highly loaded cluster, updates are therefore kept for less time. If a member is asked to retransmit an update, it can only do so if the update is still in its Sending Queue. The default (and minimum) size of this queue is 512. Each member has one sending queue.

**Receiving Queue Size**

The Receiving Queue on the cluster member keeps the updates from each cluster member until it has received a complete sequence of updates. The default (and minimum) size of this queue is 256. Each member keeps a Receiving Queue for each of the peer members.

**Synchronization Troubleshooting Options**

The following options specify the available troubleshooting options. Each option involves editing a global system configurable parameter to reconfigure the system with different value than the default.

**Enlarging the Sending Queue**

The Sending Queue on the cluster member stores locally generated sync updates. Updates in the Sending Queue are replaced by more recent updates. In a highly loaded cluster, updates are therefore kept for less time. If a member is asked to retransmit an update, it can only do so if the update is still in its Sending Queue. The default (and minimum) size of this queue is 512. Each member has one sending queue.

**To enlarge the sending queue size:**
1. Change the value of the global parameter `fw_sync_sending_queue_size`. See Advanced Cluster Configuration (on page 93).
2. You must also make sure that the required queue size survives boot. See How to Configure Gateway to Survive a Boot (on page 93).

Enlarging this queue allows the member to save more updates from other members. However, be aware that each saved update consumes memory. When changing this variable you should consider carefully the memory implications. Changes will only take effect after reboot.
Enlarging the Receiving Queue

The Receiving Queue on the cluster member keeps the updates from each cluster member until it has received a complete sequence of updates. The default (and minimum) size of this queue is 256. Each member keeps a Receiving Queue for each of the peer members.

To enlarge the receiving queue size:
1. Change the value of the global parameter `fw_syncSendingQueueSize`. See Advanced Cluster Configuration (on page 93).
2. You must also make sure that the required queue size survives boot. See How to Configure Gateway to Survive a Boot (on page 93).

Enlarging this queue means that the member can save more updates from other members. However, be aware that each saved update consumes memory. When changing this variable you should carefully consider the memory implications. Changes will only take effect after reboot.

Enlarging the Sync Timer

The `sync timer` performs sync related actions every fixed interval. By default, the sync timer interval is 100ms. The base time unit is 100ms (or 1 tick), which is therefore the minimum value.

To enlarge the sync timer:
- Change the value of the global parameter `fwhaTimerSyncRes`. See Advanced Cluster Configuration (on page 93). The value of this variable can be changed while the system is working. A reboot is not needed.

By default, `fwhaTimerSyncRes` has a value of 1, meaning that the sync timer operates every base time unit (every 100ms). If you configure this variable to n, the timer will be operated every n*100ms.

Enlarging the CPHA Timer

The `CPHA timer` performs cluster related actions every fixed interval. By default, the CPHA timer interval is 100ms. The base time unit is 100ms (or 1 tick), which is also the minimum value.

If the cluster members are geographically separated from each other, set the CPHA timer to be around 10 times the round-trip delay of the sync network.

Enlarging this value increases the time it takes to detect a failover. For example, if detecting interface failure takes 0.3 seconds, and the timer is doubled to 200ms, the time needed to detect an interface failure is doubled to 0.6 seconds.

To enlarge the CPHA timer:
- Change the value of the global parameter `fwhaTimerCphaRes`. See Advanced Cluster Configuration (on page 93). The value of this variable can be changed while the system is working. A reboot is not needed.

By default, `fwhaTimerCphaRes` has a value of 1, meaning that the CPHA timer operates every base time unit (every 100ms). If you configure this variable to n, the timer will be operated every n*100ms.

Reconfiguring the Acknowledgment Timeout

A cluster member deletes updates from its Sending Queue (described in Sending Queue Size (on page 71)) on a regular basis. This frees up space in the queue for more recent updates.

The cluster member deletes updates from this queue if it receives an ACK about the update from the peer member.

The peer member sends an ACK in one of two circumstances — on condition that the Block New Connections mechanism (described in Blocking New Connections Under Load (on page 94)) is active:
- After receiving a certain number of updates.
- If it didn't send an ACK for a certain time. This is important if the sync network has a considerable line delay, which can occur if the cluster members are geographically separated from each other.

To reconfigure the timeout after which the member sends an ACK:
- Change the value of the global parameter `fw_sync_ack_time_gap`. See Advanced Cluster Configuration (on page 93). The value of this variable can be changed while the system is working. A reboot is not needed.

  The default value for this variable is 10 ticks (10 * 100ms). Thus, if a member didn't send an ACK for a whole second, it will send an ACK for the updates it received.

**Contact Technical Support**

If the other recommendations do not help solve the problem, contact Technical Support for further assistance.

## ClusterXL Error Messages

This section lists the ClusterXL error messages. For other, less common error messages, see SecureKnowledge solution sk23642 ([http://supportcontent.checkpoint.com/solutions?id=sk23642](http://supportcontent.checkpoint.com/solutions?id=sk23642)).

### General ClusterXL Error Messages

- **FW-1: changing local mode from <mode1> to <mode2> because of ID <machine_id>**
  This log message can happen if the working mode of the cluster members is not the same, for example, if one machine is running High Availability, and another Load Sharing Multicast or Unicast mode. In this case, the internal ClusterXL mechanism tries to synchronize the configuration of the cluster members, by changing the working mode to the lowest common mode. The order of priority of the working modes (highest to lowest) is: 1. Synchronization only 2. Load Sharing 3. High Availability (Active Up) 4. High Availability (Primary Up).

- **CPHA: Received confirmations from more machines than the cluster size**
  This log message can occur during policy installation on the cluster. It means that a serious configuration problem exists in that cluster. Probably some other cluster has been configured with identical parameters and both of them have common networks.

- **fwldbcast_timer: peer X probably stopped...**
  This is caused when the member that printed this message stops hearing certain types of messages from member X. Verify that `cphaprob state` shows all members as active and that `fw ctl pstat` shows that sync is configured correctly and working properly on all members. In such a case it is fair to assume that there was a temporary connectivity problem that was fixed in the meantime. There may be several connections that may suffer from connectivity problems due to that temporary synchronization problem between the two members. On the other hand, this can indicate that the other member is really down.

- **FW-1: fwha_notify_interface: there are more than 4 IPs on interface <interface name> notifying only the first ones**
  A member of the same cluster as the reporting machine has more than three virtual IP addresses defined on the same interface. This is not a supported configuration and will harm ClusterXL functionality.

- **Sync could not start because there is no sync license**
  This is a license error message: If you have a basic Security Gateway license then sync is also licensed. Check the basic Security Gateway license using `cpli print` and `cpli check`.

- **FW-1: h_slink: an attempt to link to a link kbuf id not found**
  `fw_conn_post_inspect: fwconn_init_links failed`

  Several problems of this sort can happen during a full sync session when there are connections that are opened and closed during the full sync process. Full sync is automatic as far as possible, but it is not fully automatic for reasons of performance. A gateway continues to process traffic even when it is serving as a full sync server. This can cause some insignificant problems, such as a connection that is being deleted twice, a link to an existing link, and so forth. It should not affect connectivity or cause security issues.

- **Error SEP_IKE_owner_outbound: other cluster member packet in outbound**
Cluster in not synchronized. Usually happens in OPSEC certified third-party load sharing products for which **Support non-sticky connections** is unchecked in the cluster object **3rd Party Configuration** page.

- **FW-1: fwha_pnote_register: too many registering members, cannot register**
  The critical device (also known as Problem Notification, or pnote) mechanism can only store up to 16 different devices. An attempt to configure the 17th device (either by editing the `cphaprob.conf` file or by using the `cphaprob -d ... register` command) will result in this message.

- **FW-1: fwha_pnote_register: <NAME> already registered (# <NUMBER>)**
  Each device registered with the pnote mechanism must have a unique name. This message may happen when registering new pnote device, and means that the device `<NAME>` is already registered as with pnote number `<NUMBER>`.

- **FW-1: fwha_pnote_unregister: attempting to unregister an unregistered device <DEVICE NAME>**
  Indicates an attempt to unregister a device which is not currently registered.

- **FW-1: alert_policy_id_mismatch: failed to send a log**
  A log indicating that there is a different policy id between the two or more members was not sent. Verify all cluster members have the same policy (using `fw stat`). It is recommended to re-install the policy.

- **FW-1: fwha_receive_fwhap_msg: received incomplete HA P packet (read <number> bytes)**
  This message can be received when ClusterXL hears CCP packets of clusters of version 4.1. In that case it can be safely ignored.

**SmartView Tracker Active Mode Messages**

The following error messages can appear in SmartView Tracker Active mode. These errors indicate that some entries may not have been successfully processed, which may lead to missing synchronization information on a cluster member and inaccurate reports in SmartView Tracker.

- **FW-1: fwlddist_adjust_buf: record too big for sync. update Y for table <id> failed. fwlddist_state=<val>**
  Indicates a configuration problem on a clustered machine. Either synchronization is misconfigured, or there is a problem with transmitting packets on the sync interface. To get more information on the source of the problem:

  - Run `fw ctl pstat` (described in Monitoring Synchronization (fw ctl pstat) (on page 62)).
  - In ClusterXL clusters, run `cphaprob -a if` to get the statuses of the interfaces (see Monitoring Cluster Interfaces (on page 55)).

  To solve this problem, see Working with SmartView Tracker Active Mode (on page 95).

- **FW-1: fwldbcast_flush: active connections is currently enabled and due to high load it is making sync too slow to function properly. X active updates were dropped**
  Indicates that a clustered machine has dropped SmartView Tracker Active mode updates in order to maintain sync functionality. To solve this problem, see Working with SmartView Tracker Active Mode (on page 95).

**Sync Related Error Messages**

- **FW-1: fwldbcast_retrreq: machine <MACHINE_ID> sent a retrans request for seq <SEQ_NUM> which is no longer in my possession (current seq <SEQ_NUM>)**
  This message appears when the local member receives a retransmission request for a sequence number which in no longer in its sending window. This message can indicate a sync problem if the sending member didn't receive the requested sequence.

- **FW-1: fwlddist_save: WARNING: this member will not be fully synchronized !**
  **FW-1: fwlddist_save: current delta sync memory during full sync has reached the maximum of <MEM_SIZE> MB**
  **FW-1: fwlddist_save: it is possible to set a different limit by changing fw_sync_max_saved_buf_mem value**
  These messages may appear only during full sync. While performing full sync the delta sync updates are being saved and are applied only after the full sync process has finished. It is possible to limit the
memory used for saving delta sync updates by setting the `fw_sync_max_saved_buf_mem` variable to this limit.

- **FW-1: fwldbcast_flush: fwlddist_buf_ldbcast_unread is not being reset fast enough**
  (ur=<UNREAD_LOC>,fwlddist_buflen=<BUFFER_LEN>)
  This message may appear due to high load resulting in the sync buffer being filled faster than it is being read. A possible solution is to enlarge `fwlddist_buf_size`, as described in the Working with SmartView Tracker Active Mode (on page 95).

- **FW-1: fwlddist_mode_change: Failed to send trap requesting full sync**
  This message may appear due to a problem starting the full sync process, and indicates a severe problem. Contact Technical Support.

- **FW-1: State synchronization is in risk. Please examine your synchronization network to avoid further problems!**
  This message could appear under extremely high load, when a synchronization update was permanently lost. A synchronization update is considered to be permanently lost when it cannot be retransmitted because it is no longer in the transmit queue of the update originator. This scenario does not mean that the Security Gateway will malfunction, but rather that there is a potential problem. The potential problem is harmless if the lost sync update was to a connection that runs only on a single member as in the case of unencrypted (clear) connections (except in the case of a failover when the other member needs this update).
  The potential problem can be harmful when the lost sync update refers to a connection that is non-sticky (see Non-Sticky Connections (on page 13)), as is the case with encrypted connections. In this case the other cluster member(s) may start dropping packets relating to this connection, usually with a TCP out of state error message (see TCP Out-of-State Error Messages (on page 75)). In this case it is important to block new connections under high load, as explained in Blocking New Connections Under Load (on page 94).
  The following error message is related to this one.

- **FW-1: fwldbcast_recv: delta sync connection with member <MACHINE_ID> was lost and regained. <UPDATES_NUM> updates were lost.**
- **FW-1: fwldbcast_recv: received sequence <SEQ_NUM> (fragm <FRAG_NUM>, index <INDEX_NUM>), last processed seq <SEQ_NUM>**
  These messages appear when there was a temporary sync problem and some of the sync updates were not synchronized between the members. As a result some of the connections might not survive a failover.
  The previous error message is related to this one.

- **FW-1: The use of the non_sync_ports table is not recommended anymore. Refer to the user guide for configuring selective sync instead**
  Previous versions used a kernel table called `non_sync_ports` to implement selective sync, which is a method of choosing services that don't need to be synchronized. Selective sync can now be configured from SmartDashboard. See Choosing Services That Do Not Require Synchronization (see "Configuring Services not to Synchronize" on page 12).

**TCP Out-of-State Error Messages**

When the synchronization mechanism is under load, TCP packet out-of-state error messages may appear in the Information column of SmartView Tracker. This section explains how to resolve each error.

- **TCP packet out of state - first packet isn't SYN tcp_flags: FIN-ACK**
- **TCP packet out of state - first packet isn't SYN tcp_flags: FIN-PUSH-ACK**
  These messages occur when a FIN packet is retransmitted after deleting the connection from the connection table. To solve the problem, in SmartDashboard Global properties for Stateful Inspection, enlarge the TCP end timeout from 20 seconds to 60 seconds. If necessary, also enlarge the connection table so it won't fill completely.

- **SYN packet for established connection**
  This message occurs when a SYN is received on an established connection, and the sequence verifier is turned off. The sequence verifier is turned off for a non-sticky connection in a cluster (or in SecureXL). Some applications close connections with a RST packet (in order to reuse ports). To solve the problem, enable this behavior to specific ports or to all ports. For example, run the command:
  `fw ctl set int fw_trust_rst_on_port <port>`
Platform Specific Error Messages

**IPSO Specific Error Messages**

- **FW-1: fwha_nok_get_mc_mac_by_ip: received a NULL query**
  - FW-1: fwha_nok_get_mc_mac_by_ip: nokcl_get_clustermac returned unknown type <TYPE>
  These messages mean that automatic proxy ARP entries for static NAT configuration might not be properly installed.

- **FW-1: fwha_nokcl_sync_rx_f: received NULL mbuf from ipso. Packet dropped.**
  - FW-1: fwha_nokcl_sync_rx_f: received packet with illegal flag=<FLAG>, drop packet.
  These messages mean that an illegal CPHA packet was received and will be dropped. If this happens more than few times during boot, the cluster malfunctions.

- **FW-1: fwha_nokcl_reregister_rx: unregister old magic mac values with IPSO.**
  - FW-1: fwha_nokcl_reregister_rx: new magic mac values <MAC,FORWARD MAC> registered successfully with IPSO.
  A notification that the operation `fw ctl set int fwha_magic_mac` succeeded.

- **FW-1: fwha_nokcl_reregister_rx: error in de-registration to the sync_rx (<ERR NUM>) new magic macs values will not be applied**
  A notification that the operation `fw ctl set int fwha_magic_mac` failed. Previous MAC values will be retained.

- **FW-1: fwha_nokcl_creation_f: error in registration ...**
  - FW-1: fwha_nok_init: NOT calling nokcl_register_creation since did not de-register yet.
  - FW-1: fwha_nok_fin: failed nokcl_deregister_creation with rc=<ERROR NUM>
  These messages mean that an internal error in registration to the IPSO clustering mechanism has occurred. Verify that the IPSO version is supported by this the Security Gateway version and that the IPSO IP Clustering or VRRP cluster is configured properly.

- **FW-1: successfully (dis)connected to IPSO Clustering**
  A notification that should be normally received during Security Gateway initialization and removal.

- **FW-1: fwha_pnote_register: noksr_register_with_status failed**
  - FW-1: fwha_IPSO_pnote_expiration: mismatch between IPSO device to ckp device <DEVICE NAME>
  - FW-1: fwha_nokia_pnote_expiration: can not find the device nokia claims to be expired
  - FW-1: fwha_noksr_report_wrapper: attempting to report an unregistered device <DEVICE NAME>
  These messages may appear as a result of a problem in the interaction between the IPSO and ClusterXL device monitoring mechanisms. A reboot should solve this problem. Should this problem repeat itself contact Check Point Technical support.

### Member Fails to Start After Reboot

If a reboot (or `cpstop` followed by `cpstart`) is performed on a cluster member while the cluster is under severe load, the member may fail to start correctly. The starting member will attempt to perform a full sync with the existing active member(s) and may in the process use up all its resources and available memory. This can lead to unexpected behavior.

To overcome this problem, define the maximum amount of memory that the member may use when starting up for synchronizing its connections with the active member. By default this amount is not limited. Estimate the amount of memory required as follows:

**Table 8-10 Memory required (MB) for Full Sync.**

<table>
<thead>
<tr>
<th>New connections/second</th>
</tr>
</thead>
</table>

Which means that the Security Gateway should trust a RST coming from every port, in case a single port is not enough.
<table>
<thead>
<tr>
<th>Number of open Connections</th>
<th>100</th>
<th>1000</th>
<th>5000</th>
<th>10,000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>1.1</td>
<td>6.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10000</td>
<td>11</td>
<td>69</td>
<td>329</td>
<td></td>
</tr>
<tr>
<td>20000</td>
<td>21</td>
<td>138</td>
<td>657</td>
<td>1305</td>
</tr>
<tr>
<td>50000</td>
<td>53</td>
<td>345</td>
<td>1642</td>
<td>3264</td>
</tr>
</tbody>
</table>

**Note** - These figures were derived for cluster members using the Windows platform, with Pentium 4 processors running at 2.4 GHz.

For example, if the cluster holds 10,000 connections, and the connection rate is 1000 connections/sec you will need 69 MB for full sync.

Define the maximum amount of memory using the gateway global parameter: `fw_sync_max_saved_buf_mem`.

The units are in megabytes. For details, see Advanced Cluster Configuration (on page 93).
Chapter 9

ClusterXL Advanced Configuration

In This Chapter

Working with VPNs and Clusters
Working with NAT and Clusters
Working with VLANS and Clusters
Monitoring the Interface Link State
Link Aggregation and Clusters
Advanced Cluster Configuration
Defining Disconnected Interfaces
Configuring Policy Update Timeout
Enhanced 3-Way TCP Handshake Enforcement
Configuring Cluster Addresses on Different Subnets
Moving from a Single Gateway to a ClusterXL Cluster
Adding Another Member to an Existing Cluster
Configuring ISP Redundancy on a Cluster
Enabling Dynamic Routing Protocols in a Cluster Deployment

Working with VPNs and Clusters

Configuring VPN and Clusters

Configuring a Security Gateway cluster using SmartDashboard is very similar to configuring a single Security Gateway. All attributes of the VPN are defined in the Gateway Cluster object, except for two attributes that are defined per cluster member.

1. Go to the Gateway Cluster Properties window, Cluster Members page. For each cluster member, in the Cluster member Properties window, configure the VPN tab:
   - **Office Mode for Remote access** — If you wish to use Office Mode for remote access, define the IP pool allocated to each cluster member.
   - **Hardware Certificate Storage List** — If your cluster member supports hardware storage for IKE certificates, define the certificate properties. In that case, Security Management server directs the cluster member to create the keys and supply only the required material for creation of the certificate request. The certificate is downloaded to the cluster member during policy installation.

2. In a VPN cluster, IKE keys are synchronized. In the Synchronization page of the Gateway Cluster Properties window, make sure that Use State Synchronization is selected, even for High Availability configurations.

3. In the Topology page of the Gateway Cluster Properties window, define the encryption domain of the cluster. Under VPN Domain, choose one of the two possible settings:
   - **All IP addresses behind cluster members based on topology information**. This is the default option.
   - **Manually Defined**. Use this option if the cluster IP address is not on the member network, in other words, if the cluster virtual IP address is on a different subnet than the cluster member interfaces. In that case, select a network or group of networks, which must include the virtual IP address of the cluster, and the network or group of networks behind the cluster.
Defining VPN Peer Clusters with Separate Security Management Servers

When working with a VPN peer that is a Check Point Gateway cluster, and the VPN peer is managed by a different Security Management server, do NOT define another cluster object. Instead, do the following:

1. In the objects tree, Network Objects branch, right click and select New Check Point Externally Managed Gateway.
2. In the Topology page, add the external and internal cluster interface addresses of the VPN peer. Do not use the cluster member interface addresses, except in the following cases:
   - If the external cluster is of version 4.1, add the IP addresses of the cluster member interfaces.
   - If the cluster is an OPSEC certified product (excluding IPSO), you may need to add the IP addresses of the cluster members.

   When adding cluster member interface IP addresses, in the interface Topology tab, define the interface as Internal, and the IP Addresses behind this interface as Not defined.
3. In the VPN Domain section of the page, define the encryption domain of the externally managed gateway to be behind the internal virtual IP address of the gateway. If the encryption domain is just one subnet, choose All IP addresses behind cluster members based on topology information. If the encryption domain includes more than one subnet, it must be Manually Defined.

Working with NAT and Clusters

Cluster Fold and Cluster Hide

Network Address Translation (NAT) is a fundamental aspect of the way ClusterXL works.

When a cluster member establishes an outgoing connection towards the Internet, the source address in the outgoing packets, is the physical IP address of the cluster member interface. The source IP address is changed using NAT to that of the external virtual IP address of the cluster. This address translation is called "Cluster Hide".

For OPSEC certified clustering products, this corresponds to the default setting in the 3rd Party Configuration page of the cluster object, of Hide Cluster Members' outgoing traffic behind the Cluster's IP address being checked.

When a client establishes an incoming connection to external (virtual) address of the cluster, ClusterXL changes the destination IP address using NAT to that of the physical external address of one of the cluster members. This address translation is called "Cluster Fold".

For OPSEC certified clustering products, this corresponds to the default setting in the 3rd Party Configuration page of the cluster object, of Forward Cluster's incoming traffic to Cluster Members' IP addresses being checked.

Configuring NAT on the Gateway Cluster

Network Address Translation (NAT) can be performed on a Gateway Cluster, in the same way as it is performed on a Gateway. This NAT is in addition to the automatic "Cluster Fold" and "Cluster Hide" address translations.

To configure NAT, edit the Gateway Cluster object, and in the Gateway Cluster Properties window, select the NAT page. Do NOT configure the NAT tab of the cluster member object.

Configuring NAT on a Cluster Member

It is possible to perform Network Address Translation (NAT) on a non-cluster interface of a cluster member. A possible scenario for this is if the non-Cluster interface of the cluster member is connected to another (non-cluster) internal Security Gateway, and you wish to hide the address of the non-Cluster interface of the cluster member.
Performing this NAT means that when a packet originates behind or on the non-Cluster interface of the cluster member, and is sent to a host on the other side of the internal Security Gateway, the source address of the packet will be translated.

**To configure NAT on a non-cluster interface of a cluster member gateway:**

1. Edit the Gateway Cluster object.
2. In the Cluster Member page of the Gateway Cluster Properties window, edit the Cluster Member object.
3. In the Cluster Member Properties window, click the NAT tab.
4. Configure Static or Hide NAT as desired.

---

**Working with VLANS and Clusters**

**VLAN Support in ClusterXL**

A VLAN switch tags packets that originate in a VLAN with a four-byte header that specifies which switch port it came from. No packet is allowed to go from a switch port in one VLAN to a switch port in another VLAN, apart from ports ("global" ports) that are defined so that they belong to all the VLANs.

The cluster member is connected to the global port of the VLAN switch, and this logically divides a single physical port into many VLAN ports each associated with a VLAN tagged interface (VLAN interface) on the cluster member.

When defining VLAN tags on an interface, cluster IP addresses can be defined only on the VLAN interfaces (the tagged interfaces). Defining a cluster IP address on a physical interface that has VLANs is not supported. This physical interface has to be defined with the Network Objective Monitored Private.

**Note** - ClusterXL does not support VLANS on Windows 2000 or Windows 2003 Server.

**Connecting Several Clusters on the Same VLAN**

It is not recommended to connect the non-secured interfaces (the internal or external cluster interfaces, for example) of multiple clusters to the same VLAN. A separate VLAN, and/or switch is needed for each cluster.

Connecting the secured interfaces (the synchronization interfaces) of multiple clusters is also not recommended for the same reason. Therefore, it is best to connect the secured interfaces of a given cluster via a crossover link when possible, or to an isolated VLAN.

If there is a need to connect the secured or the non-secured interfaces of multiple clusters to the same VLAN you need to make changes to:

- The destination MAC address, to enable communication between the cluster and machines outside the cluster (for ClusterXL Load Sharing Multicast Mode clusters only).
- The source MAC address of the cluster, to enable Cluster Control Protocol communication between cluster members.

**Changes to the Destination MAC Address**

This section applies to ClusterXL Load Sharing Multicast Mode only.

**How the Destination Cluster MAC Address is Assigned in Load Sharing Multicast Mode**

When a machine that is outside the cluster wishes to communicate with the cluster, it sends an ARP query with the cluster (virtual) IP address. The cluster replies to the ARP request with a multicast MAC address, even though the IP address is a unicast address.

This destination multicast MAC address of the cluster is based on the unicast IP address of the cluster. The upper three bytes are **01.00.5E**, and they identify a Multicast MAC in the standard way. The lower three...
bytes are the same as the lower three bytes of the IP address. An example MAC address based on the IP address 10.0.10.11 is shown below.

**Figure 9-7** The Multicast MAC Address of a Cluster

![Multicast MAC Address Diagram]

Duplicate Multicast MAC Addresses: The Problem

When more than one cluster is connected to the same VLAN, the last three bytes of the IP addresses of the cluster interfaces connected to the VLAN must be different. If they are the same, then communication from outside the cluster that is intended for one of the clusters will reach both clusters, which will cause communication problems.

For example, it is OK for the cluster interface of one of the clusters connected to the VLAN to have the address 10.0.10.11, and the cluster interface of a second cluster to have the address 10.0.10.12. However, the following addresses for the interfaces of the first and second clusters will cause complications: 10.0.10.11 and 20.0.10.11.

Duplicate Multicast MAC Addresses: The Solution

The best solution is to change the last three bytes of the IP address of all but one of the cluster interfaces that share the same last three bytes of their IP address.

If the IP address of the cluster interface cannot be changed, you must change the automatically assigned multicast MAC address of all but one of the clusters and replace it with a user-defined multicast MAC address. Proceed as follows:

1. In the ClusterXL page of the cluster object, select Load Sharing>Multicast Mode. In the Topology tab, edit the cluster interface that is connected to same VLAN as the other cluster.
2. In the Interface Properties window, General tab, click Advanced.
3. Change the default MAC address, and carefully type the new user defined MAC address. It must be of the form 01:00:5e:xy:yy:yy where x is between 0 and 7 and y is between 0 and f(hex).

Changes to the Source MAC Address

This section applies to all ClusterXL modes, both High Availability and Load Sharing, and to OPSEC certified clustering products.

How the Source Cluster MAC Address is Assigned

Cluster members communicate with each other using the Cluster Control Protocol (CCP). CCP packets are distinguished from ordinary network traffic by giving CCP packets 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. Its value indicates its purpose:

<table>
<thead>
<tr>
<th>Default value of fifth byte</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xfe</td>
<td>CCP traffic</td>
</tr>
<tr>
<td>0xfd</td>
<td>Forwarding layer traffic</td>
</tr>
</tbody>
</table>

- The sixth byte is the ID of the sending cluster member
Duplicate Source Cluster MAC Addresses: The Problem

When more than one cluster is connected to the same VLAN, if CCP and Forwarding Layer traffic uses Multicast MAC address for the destination, this traffic reaches only the intended cluster. However, if Broadcast MAC address is used for Destination for CCP and for Forwarding Layer traffic (and in certain other cases), cluster traffic intended for one cluster is seen by all connected clusters. If this traffic is processed by the wrong cluster, it will cause communication problems.

Duplicate Source Cluster MAC Addresses: The Solution

To ensure that the source MAC address in packets from different clusters that are connected to the same VLAN can be distinguished, change the MAC source address of the cluster interface that is connected to the VLAN in all but one of the clusters.

Use the following gateway configuration parameters to set more than one cluster on the same VLAN. These parameters apply to both ClusterXL and OPSEC certified clustering products.

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

Changing the values of these gateway configuration parameters alters the fifth part of the source MAC address of Cluster Control Protocol and forwarded packets. Use any value as long as the two gateway configuration parameters are different. To avoid confusion, do not use the value 0x00.

For instruction about how to change these parameters, see How to Configure Gateway Configuration Parameters (on page 93).

Monitoring the Interface Link State

Enabling Interface Link State Monitoring shortens the time it takes for ClusterXL to detect an interface failure. By monitoring the link state (i.e. the electrical state) of an interface, ClusterXL is immediately alerted to connectivity issues concerning a certain network interface, such as a disconnected cable, or an electrical failure (real or simulated) on a switch.

Interface Link State Monitoring requires an interface device driver that supports link state detection. The device driver reports the link state as either connected or disconnected.

Monitoring the interface link state is particularly useful in scenarios where a monitored interface (either a cluster interface or a monitored private interface) sends ICMP ECHO probe requests which are not answered by hosts or routers on the connected subnet.

When enabled, ClusterXL immediately detects when an interface goes down. When disabled, ClusterXL determines whether an interface is malfunctioning by watching subsecond timeout expiration.

Monitoring Interface Link State is disabled by default.

Note - Interface Link State Monitoring requires an interface device driver that supports link state detection, and is supported on Linux and SecurePlatform only. This feature only works in two-member ClusterXL clusters where two interfaces are connected directly with a cross-cable. See sk31336 (http://supportcontent.checkpoint.com/solutions?id=sk31336).
Enabling Interface Link State Monitoring

To enable (or disable) Interface Link State Monitoring:

- Set the global parameter `fwha_monitor_if_link_state`.

Usage:

```
fw ctl set int fwha_monitor_if_link_state <0|1>
```

Options:
- 0 – disables Interface Link State Monitoring. This is the default setting.
- 1 – enables Interface Link State Monitoring

For instructions on how to make these configuration parameters survive reboot, see SecureKnowledge sk26202 [http://supportcontent.checkpoint.com/solutions?id=sk26202].

Link Aggregation and Clusters

In This Section

- Overview 83
- Link Aggregation - High Availability Mode 84
- Link Aggregation - Load Sharing Mode 87
- Defining VLANs on an Interface Bond 89
- Performance Guidelines for Link Aggregation 89
- ClusterXL Commands for Interface Bonds 90
- Troubleshooting Bonded Interfaces 92

Overview

Link Aggregation (also known as NIC teaming) involves bonding multiple network interfaces on a Security Gateway. The interface bond supplies high availability in case of interface failure and, in Load Sharing mode, can significantly increase total throughput.

Note - Link Aggregation is supported only on Check Point SecurePlatform.

In an interface bond, between two and eight interfaces are set to act as a single interface, using the same IP address.

The bond is a virtual interface, defined in SecurePlatform similarly to a physical interface. Each physical interface in a bond is called a slave of that bond. Enslaved interfaces do not function independently of the bond.

Link Aggregation can be configured to one of two modes:

- **High Availability (Active/Backup) mode** - only one interface at a time is active. Upon interface failure, the bond fails over to another interface. Different interfaces of the bond can be connected to different switches, to add switch high availability to interface high availability.
  
  Note - Link-state initiated internal bond failover requires a network interface that supports the Media-Independent Interface (MII) standard.

- **Load Sharing (Active/Active) mode** - all interfaces are active, for different connections. Connections are balanced between interfaces according to network layers three and four, and follow either the IEEE
802.3ad standard or XOR. Load Sharing mode has the advantage of increasing throughput, but requires connecting all the interfaces of the bond to one switch.

For Link Aggregation High Availability mode and for Link Aggregation Load Sharing mode:

- The number of bond interfaces that can be defined is limited by the maximum number of interfaces supported by each platform. See the Release Notes for the appropriate Check Point release.
- Up to 8 NICs can be configured in a single High Availability or Load Sharing bond.

**Link Aggregation - High Availability Mode**

In This Section

- Fully Meshed Redundancy 85
- Bond Failover 85
- Creating an Interface Bond in High Availability Mode 85
- Failover Support for VLANs 86

When dealing with mission-critical applications, an enterprise requires its network to be highly available. Clustering provides redundancy, and thus, high availability, at the gateway level. Without Link Aggregation, redundancy of Network Interface Cards (NICs) or of the switches on either side of the gateway are only possible in a cluster, and only by failover of the gateway to another cluster member.

You can have redundancy of gateway clustering without Link Aggregation. If a switch or gateway fails, a High Availability cluster solution provides system redundancy. For example, you can have a redundant system with two synchronized Security Gateway cluster members deployed in a redundant topology.

**Figure 9-8** Topology without Interface Bonding

In this scenario:

- GW-1 and GW-2 are cluster members. Each has one external Network Interface Card (NIC) connected to an external switch (S-1 and S-2, respectively).
- S-1 and S-2 are switches
- C-1 and C-2 are interconnecting networks

If Member 1, its NIC, or S-1 fails, Member 2 becomes the only active gateway, connecting to switch S-2 over network C-2. If any component fails (gateway, NIC, or switch), the result of the failover is that no further redundancy exists. A further failure of any active component completely stops network traffic.

Link Aggregation provides high availability of NICs. If one fails, the other can function in its place. This functionality is in High Availability mode and in Load Sharing mode.
**Fully Meshed Redundancy**

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.

A fully meshed topology further enhances the redundancy in the system by providing a backup to both the interface and the switch, essentially backing up the cable. Each cluster member has two external interfaces, one connected to each switch.

**Figure 9-9** Fully meshed topology with Interface Bonding

In this scenario:
- Member-1 and Member-2 are cluster members in the High Availability mode, each connected to the two external switches.
- S-1 and S-2 are switches
- C-1, C-2, C-3 and C-4 are network connections

**Bond Failover**

In Link Aggregation High Availability mode, when the gateway is part of a cluster, bond internal failover can occur in one of these cases:

- An active interface detects a failure in the link state, in a monitored interface.
- ClusterXL detects a failure in sending or receiving Cluster Control Protocol (CCP) keep-alive packets.

Either of these failures will induce a failover within the interface bond, or between cluster members, depending on the circumstances. The section below describes the two types of failover processes.

When a failure is detected, a log is recorded. You can see it in SmartView Tracker.

**Creating an Interface Bond in High Availability Mode**

Do these procedures to create an interface bond:

- Removing IP Addresses from Slave Interfaces 86
- Setting Slave Interfaces as Disconnected 86
- Defining the Interface Bond 86
- Verifying that the Bond is Functioning Properly 86
Removing IP Addresses from Slave Interfaces

Before you define an interface bond, make sure the slave (physical) interfaces do not have IP addresses:

1. Start the SecurePlatform configuration utility:
   ```bash
   sysconfig
   ```
2. Select Network Connections.
3. For each interface to be enslaved:
   a) Select Configure connection.
   b) Select the relevant physical interface.
   c) Select Remove IP from interface.
   d) Return to Network Connections.
4. Exit the SecurePlatform configuration utility.

Setting Slave Interfaces as Disconnected

Disconnected interfaces are cluster member interfaces that are not monitored by the ClusterXL mechanism. If a disconnected interface fails, failover does not occur.

To define a slave interface as disconnected in SecurePlatform:

1. In `$FWDIR/conf/` create a file with this name: `discntd.if`
2. On separate lines in the file, enter the name of each physical interface that will function as a slave/bond pair.

Defining the Interface Bond

When the slave interfaces are without IP addresses, define the bond:

1. Start the SecurePlatform configuration utility:
   ```bash
   sysconfig
   ```
2. Select Network Connections.
3. Select Add new connection.
4. Select Bond.
5. For each interface to be enslaved under the bond, type its number in the list, and press Enter.
6. Enter `n` to go to the next step.
7. Select High Availability.
8. Choose whether to use default parameters (recommended) or to customize them.
9. Choose whether to set a primary slave interface, or not (recommended).
   A primary slave interface, after failing and coming back up, automatically returns to Active status, even if failover to the other interface occurred. If there is no primary interface, failover causes the other interface to become active and remain so until it fails.
10. Define the IP address and network mask of the new interface bond.
11. Exit the SecurePlatform configuration utility.

Verifying that the Bond is Functioning Properly

After installation or failover, it is recommended to verify that the bond is up, by displaying bond information.

1. Run:
   ```bash
   cphaprob -a if
   ```
   Make sure that the bond status is reported as UP.
2. Run:
   ```bash
   cphaconf show_bond <bond name>
   ```
   Check that the bond is correctly configured.
**Failover Support for VLANs**

In Link Aggregation High Availability mode, ClusterXL monitors VLAN IDs for connectivity failure or miscommunication, and initiate a failover when a failure is detected.

In a VLAN-enabled switched environment, ClusterXL monitors the VLAN with the lowest ID number. The monitoring is conducted by sending ClusterXL Control Protocol (CCP) packets on round-trip paths at a set interval. The lowest VLAN ID indicates the status of the physical connection. This VLAN ID is always monitored, and a connectivity failure causes ClusterXL to initiate a failover. ClusterXL will not detect a VLAN configuration problem on the switch.

**Figure 9-10** Interface Bonding with VLANS

**Link Aggregation - Load Sharing Mode**

In This Section

- Workflow of Interface Bond in Load Sharing Mode 88
- Configuring Cisco Switches for Load Sharing 88

In Load Sharing mode, Link Aggregation supplies load sharing, in addition to High Availability. All slave interfaces are active, and connections are balanced between the bond's slave interfaces, similar to the way ClusterXL balances connections between cluster members.

In Load Sharing mode, each connection is assigned to a specific slave interface. For the individual connection, only one slave interface is active. On failure of that interface, the bond does failover of the connection to one of the other interfaces, which adds the failed interface's connection to the connections it is already handling.

Connections are balanced between slave interfaces according to network layers three and four, and follow one of these standards:

- 802.3ad - includes LACP and is the recommended mode, but some switches may not support this mode.
- XOR.

In Load Sharing mode, all the interfaces of a bond must be connected to the same switch. The switch itself must support and be configured for Link Aggregation, by the same standard (802.3ad or XOR) as the gateway bond.

Load Sharing needs Performance Pack to be running.
Workflow of Interface Bond in Load Sharing Mode

Creating a Load Sharing bond is similar to creating a High Availability bond. The procedures for removing IP addresses from slaves, disconnecting slave interfaces, and verifying the bond are the same.

To create a Load Sharing bond:
1. Make sure the switches are configured for the standard you are using (802.3ad or XOR).
2. Removing IP Addresses from Slave Interfaces (on page 86)
3. Setting Slave Interfaces as Disconnected (on page 86)
4. Defining Interface Bond in Load Sharing Mode
5. Setting Critical Required Interfaces
6. Verifying that the Bond is Functioning Properly (on page 86)

Defining Interface Bond in Load Sharing Mode

To define the interface bond:
1. Start the SecurePlatform configuration utility:
   sysconfig
2. Select Network Connections.
3. Select Add new connection.
4. Select Bond.
5. For each interface to be enslaved under the bond, type its number in the list, and press Enter.
6. Enter n to go to the next step.
7. Select Load Sharing.
8. Choose the Load Sharing standard: 802.3ad or XOR.
9. Choose whether to use default parameters (recommended) or to customize them.
10. Define the IP address and network mask of the new interface bond.
11. Exit the SecurePlatform configuration utility.

Setting Critical Required Interfaces

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.
Configuring Cisco Switches for LoadSharing

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-if)#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-if)#interface port-channel 1
Switch(config-if)#switchport access vlan <the wanted vlan number>
Switch(config-if)#end
Switch#write

Defining VLANs on an Interface Bond

VLANs can be defined on an interface bond in the same way as on a regular interface.

To define a VLAN on an interface bond:
1. Start the SecurePlatform configuration utility:
   sysconfig
2. Select Network Connections.
3. Select Add new connection.
4. Select VLAN.
5. Select the interface or interface bond on which to define the VLAN.
6. Enter a VLAN ID.
7. Define the IP addresses for the VLAN.
8. Exit the SecurePlatform configuration utility.

Performance Guidelines for Link Aggregation

To get the best performance, use static affinity for Link Aggregation.

Setting Affinities

If you are running Performance Pack in a multi-core system, after you define bonds, set affinities manually. Use the -s option of the sim affinity command (see Performance Pack documentation).

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:

<table>
<thead>
<tr>
<th>bond0</th>
<th>bond1</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth0</td>
<td>eth3</td>
</tr>
<tr>
<td>eth1</td>
<td>eth4</td>
</tr>
<tr>
<td>eth2</td>
<td>eth5</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>bond0</th>
<th>bond1</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth0</td>
<td>core 0</td>
</tr>
<tr>
<td>eth1</td>
<td>core 1</td>
</tr>
<tr>
<td>eth2</td>
<td>core 2</td>
</tr>
<tr>
<td>eth3</td>
<td>core 0</td>
</tr>
<tr>
<td>eth4</td>
<td>core 1</td>
</tr>
<tr>
<td>eth5</td>
<td>core 3</td>
</tr>
</tbody>
</table>

### ClusterXL Commands for Interface Bonds

<table>
<thead>
<tr>
<th>cphaconf show_bond</th>
<th>See status of one interface bond or summary of all bonds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>cphaconf show_bond [ &lt;bond-name&gt;</td>
</tr>
<tr>
<td>Options</td>
<td>Parameter</td>
</tr>
<tr>
<td></td>
<td>bond-name</td>
</tr>
<tr>
<td></td>
<td>-a</td>
</tr>
</tbody>
</table>
**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
--------------------------------------------
ed2 | Active | Yes  
ed3 | Active | Yes  
ed4 | Active | Yes  
ed5 | Active | Yes

**Comments**

The report results show:

- Required slave interfaces (*Setting Critical Required Interfaces* on page 88)

- Status value:
  - **Down** - (Load Sharing only) the physical link is down.
  - **Active** - currently handling traffic.
  - **Standby** - (High Availability only) the interface is ready and can support internal bond failover.
  - **Not Available** - (High Availability only) the physical link is broken, or the Cluster member is in status *down*. The bond cannot failover in this state.

- **Link** - if the physical link exists.

**cphaconf failover_bond**  
Starts interface bond internal failover (High Availability only)

**Syntax**

cphaconf failover_bond <bond-name>

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bond-name</td>
<td>name of target bond</td>
</tr>
</tbody>
</table>

**chaprob -a if**  
Displays status of all interface bonds and VLANs

**Syntax**

cphaprob -a if

[Expert@GW-1]# cphaprob -a if

Required interfaces: 5  
bond0 UP non sync(non secured), broadcast, bond, can failover  
bond2 UP sync(secured), multicast, bond Load Sharing  
bond1 UP non sync(non secured), multicast, bond Load Sharing  
Virtual cluster interfaces: 4  
bond0 192.168.34.60  
bond1.60 10.34.60.1  
bond1.61 10.34.61.1  
bond1.62 10.34.62.1

**Comments**

Use this command to see if a High Availability bond can failover.
Troubleshooting Bonded Interfaces

In This Section

Troubleshooting Workflow 92
Connectivity Delays on Switches 92

Troubleshooting Workflow

1. Check the status of the bond ("Verifying that the Bond is Functioning Properly" on page 86).
2. If there is a problem, see if the physical link is down:
   a) Run:
      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. See if a cluster member is down:
   cphaprob state
   If any of the cluster members have a Firewall State other than active, continue with the cphaprob state troubleshooting ("The cphaprob Command" on page 52).
4. View the logs 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. These are suggestions for reducing the startup time after link failure.

1. Disable autonegotiation on the relevant interface.
2. On some Cisco switches, enable the PortFast feature.
3. Disable STP on the ports.

Warnings 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 crash.

Sample Configuration of PortFast Feature on a Cisco Switch

The following are the commands necessary to enable PortFast on a GigabitEthernet 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
Advanced Cluster Configuration

How to Configure Gateway Configuration Parameters

A number of synchronization and ClusterXL capabilities are controlled by means of Security Gateway configuration parameters. Run these commands on the Security Gateway as follows:

```
fw ctl set int Parameter <value>
```

*Parameter* is any of the parameters described in the following sections.

Changes to their default values must be implemented on all cluster members. Setting different values on cluster members can cause configuration problems and possibly connection failures.

All these gateway configuration parameters can be configured to survive a boot. The way to do this varies with the operating system.

How to Configure Gateway to Survive a Boot

Gateway configuration parameters that are changed using the `fw ctl set int` command do not survive reboot. The way to do make them survive a reboot varies with the operating system. In the following instructions, *Parameter* is any of the parameters described in the following sections.

**Linux/SecurePlatform**

1. Edit the file `$FWDIR/boot/modules/fwkern.conf`.
2. Add the line `Parameter=<value in hex>`.
3. Reboot.

**Windows**

1. Edit the registry.
2. Add a DWORD value named `Parameter` under the key `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\FW1\Parameters\Globals`.
3. Reboot.

Setting Module Variables in IPSO 6.1 and Later

When you install IPSO or run Voyager for the first time on a new platform, the Firewall Kernel Tuning Configuration page does not appear. If a customer service representative instructs you to use this page, you must first display it by performing these steps:

1. Establish a command line connection to the platform (using a network connection or console connection).
2. At the IPSO shell prompt, enter `nokia[admin]# dbset advanced:loadert`.
3. Run Voyager (or exit Voyager and run it again if Voyager was open when you entered the previous command).
4. Click **Configuration > Tools > Firewall Kernel Tuning** in the navigation tree.
5. Configure the variables as instructed by support and click **Apply**. Clicking Apply applies the firewall kernel variables and also saves the Voyager configuration so that the Firewall Kernel Tuning Configuration page will appear again if you reboot the platform.

Controlling the Clustering and Synchronization Timers

The following gateway configuration parameters are used to control the clustering and synchronization timers. Changing the default values is not recommended.
### Clustering and Synchronization timers

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>fwha_timer_cpha_res</code></td>
<td>The frequency of ClusterXL operations on the cluster.</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Operations occur every: 10 multiplied by <code>fwha_timer_cpha_res</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>multiplied by <code>fwha_timer_base_res</code> milliseconds</td>
<td></td>
</tr>
<tr>
<td><code>fwha_timer_sync_res</code></td>
<td>The frequency of sync flush operations on the cluster.</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Operations occur every: 10 multiplied by <code>fwha_timer_sync_res</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>multiplied by <code>fwha_timer_base_res</code> milliseconds</td>
<td></td>
</tr>
<tr>
<td><code>fwha_timer_base_res</code></td>
<td>Must be divisible by 10 with no remainders.</td>
<td>10</td>
</tr>
</tbody>
</table>

### Blocking New Connections Under Load

The reason for blocking new connections is that new connections are the main source of new synchronization traffic, and synchronization may be put at risk if new traffic continues to be processed at this rate.

A related error message is: "FW-1: State synchronization is in risk. Please examine your synchronization network to avoid further problems!" on page 163.

Reducing the amount of traffic passing through the Security Gateway protects the synchronization mechanism.

- `fw_sync_block_new_conns` allows Security Gateway to detect heavy loads and start blocking new connections. Load is considered heavy when the synchronization transmit queue of the firewall starts to fill beyond the `fw_sync_buffer_threshold`.

  - To enable load detection, set to 0.
  - To disable load detection, set to -1 (0xFFFFFFFF hex) (default value).

  Note that blocking new connections when sync is busy is only recommended for Load Sharing ClusterXL deployments. While it is possible to block new connections in High Availability mode, doing so does not solve inconsistencies in sync, as High Availability mode precludes that from happening. This parameter can be set to survive boot using the mechanism described in How to Configure Gateway to Survive a Boot (on page 93).

- `fw_sync_buffer_threshold` is the maximum percentage of the buffer that may be filled before new connections are blocked. By default it is set to 80, with a buffer size of 512. By default, if more than 410 consecutive packets are sent without getting an ACK on any one of them, new connections are dropped. When blocking starts, `fw_sync_block_new_conns` is set to 1. When the situation stabilizes it is set back to 0.

- `fw_sync_allowed_protocols` is used to determine the type of connections that can be opened while the system is in a blocking state. Thus, the user can have better control over the system's behavior in cases of unusual load. The `fw_sync_allowed_protocols` variable is a combination of flags, each specifying a different type of connection. The required value of the variable is the result of adding the separate values of these flags. For example, the default value of this variable is 24, which is the sum of `TCP_DATA_CONN_ALLOWED` (8) and `UDP_DATA_CONN_ALLOWED` (16), meaning that the default allows only TCP and UDP data connections to be opened under load.
Working with SmartView Tracker Active Mode

Active mode in SmartView Tracker shows connections currently open through any of the Security Gateways that are sending logs to the currently active Log File on the Security Management server.

Active mode tends to slow down synchronization. If that happens, the synchronization mechanism randomly drops Active connection updates in order to maintain synchronization. The drop will be accompanied by one of the error messages described in SmartView Tracker Active Mode Messages.

Active mode view is not recommended on a heavily loaded cluster. To obtain a more accurate report of Active connections under load, two solutions are available. They apply both to a cluster and to a single Security Gateway:

1. Enlarge `fwldist_buf_size`
   The `fwldist_buf_size` parameter controls the size of the synchronization buffer in words. (Words are used for both synchronization and in SmartView Tracker Active mode. 1 word equals 4 Bytes). The default is 16k words. The maximum value is 64k words and the minimum value is 2k words.
   If changing this parameter, make sure that it survives boot, because the change is only applied after a reboot. Use the mechanism described in How to Configure Gateway Configuration Parameters (on page 93).

2. Obtain a Hotfix from Technical Support
   Obtain a Check Point Technical Support Hotfix. This Hotfix has a variable that controls the rate at which Active connections are read by `fwd` on the gateway before being sent to the Security Management server. Note that this solution requires additional CPU resources.

Reducing the Number of Pending Packets

ClusterXL prevents out-of-state packets in non-sticky connections. It does this by holding packets until a Sync ACK is received from all other active cluster members. If for some reason a Sync ACK is not received, the Security Gateway on the cluster member will not release the packet, and the connection will not be established.

To find out if held packets are not being released, run the `fw ctl pstat` command. If the output of the command shows that the Number of Pending Packets is large under normal loads (more than 100 pending packets), and this value does not decrease over time, use the `fwldbcast_pending_timeout` parameter to reduce the number of pending packets.

Change the value of `fwldbcast_pending_timeout` from the default value of 50 to a value lower than 50.
The value is in ticks units, where each tick is equal to 0.1 sec, so that 50 ticks is 5 seconds.
The value represents the time after which packets are released even if Sync ACKs are not received.

Configuring Full Synchronization Advanced Options

When a cluster member comes up after being rebooted (or after `cpstart`), it has to perform Full Synchronization. As a first step in the Full Synchronization process, it performs a handshake with one of the other active cluster members. Only if this handshake succeeds does the cluster member continue with the Full Synchronization process.
The extended handshake that takes place (by default) exchanges information between cluster members. This information includes version information, information about the installed Check Point products, and can include information about which the VPN kernel tables are currently active. The extended handshake is unrelated to the exchange of kernel table information that happens later in the Full Synchronization.

All cluster members must have the same Check Point products and versions installed. The extended handshake identifies when different products are installed on the cluster members. When different products are installed, a console warning and a log message are issued.

In order to support backward compatibility, it is possible to change the behavior of the extended handshake by means of the following Gateway Configuration Parameters. How to edit these parameters is explained in Advanced Cluster Configuration (on page 93):

- `fw_sync_no ld_trans` has the default value of 1. Set to 0 in order to exchange kernel table information between members in the first phase of the Full Synchronization process.
- `fw_sync_no_conn_trans` has the default value of 0. Set to 1 in order not to exchange installed product information between members in the first phase of the Full Synchronization process.
- `fw_sync_fcu_ver_check` has the default value of 1. Set to 0 to allow Full Connectivity Upgrade for versions that do not comply with the version requirements.

(http://supportcontent.checkpoint.com/documentation_download?ID=11648)

### Defining Disconnected Interfaces

Disconnected interfaces are cluster member interfaces that are not monitored by the ClusterXL mechanism. You may wish to define an interface as disconnected if the interface is down for a long time, and you wish the cluster member to continue to be active.

The processes listed below are equivalent to defining a non-monitored interface from the Topology page, with the exception that the GUI method works only for interfaces that have a defined IP address.

#### Defining a Disconnected Interface on Unix

Create a file under `$FWDIR/conf/discntd.if` and write the name of each interface that you do not want monitored by ClusterXL on a separate line.

#### Defining a Disconnected Interface on Windows

1. Open the `regedit32` registry editor. Do not use `regedit`.
2. Under `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\CPHA` create a new value with the following characteristics:
   - **Value Name**: `DisconnectedInterfaces`
   - **Data Type**: REG_MULTI_SZ
3. Add the interface name. To obtain the interface system name run the command:
   - `fw getifs`
4. Add this name to the list of disconnected interfaces using the following format:
   - `\device<System Interface Name>`
5. Run `cphastop` and then `cphastart` to apply the change.

#### Configuring Policy Update Timeout

When policy is installed on a Gateway Cluster, the cluster members undertake a negotiation process to make sure all of them have received the same policy before they actually apply it. This negotiation process has a timeout mechanism which makes sure a cluster member does not wait indefinitely for responses from other cluster members, which is useful in cases when another cluster member goes down when policy is being installed (for example).
In configurations on which policy installation takes a long time (usually caused by a policy with a large number of rules), a cluster with more than two machines, and slow machines, this timeout mechanism may expire prematurely.

It is possible to tune the timeout by setting the following parameter:

```
fwha_policy_update_timeout_factor.
```

The default value is 1 which should be sufficient for most configurations. For configurations where the situation described above occurs, setting this parameter to 2 should be sufficient. Do NOT set this parameter to a value larger than 3.

Enhanced 3-Way TCP Handshake Enforcement

The standard enforcement for a 3-way handshake that initiates a TCP connection provides adequate security by guaranteeing one-directional stickiness. This means that it ensures that the SYN-ACK will always arrive after the SYN. However, it does not guarantee that the ACK will always arrive after the SYN-ACK, or that the first data packet will arrive after the ACK.

If you wish to have stricter policy that denies all out-of-state packets, you can configure the synchronization mechanism so that all the TCP connection initiation packets arrive in the right sequence (SYN, SYN-ACK, ACK, followed by the data). The price for this extra security is a considerable delay in connection establishment.

To enable enhanced enforcement, use the Database Tool to change the `sync_tcp_handshake_mode` property from `minimal_sync` (default value) to `complete_sync`.

Configuring Cluster Addresses on Different Subnets

Introduction to Cluster Addresses on Different Subnets

Cluster IPs are virtual IP addresses given to ClusterXL objects, which differ from the unique IPs of the individual cluster machines. These addresses enable the cluster to be seen as a single gateway, thus allowing it to serve as a router in a network that is unaware of the cluster's internal structure and status.

In previous versions, cluster IP addresses had to be configured on the same subnets as those used by the unique addresses of the cluster members. Cluster IPs can reside on subnets other than those of the members. The advantage of this is that it

- Enables a multi-machine cluster to replace a single-machine gateway in a pre-configured network, without the need to allocate new addresses to the cluster members.
- Makes it possible to use only one routable address for the ClusterXL Gateway Cluster

**Note** - This capability is available only for ClusterXL Gateway Clusters. For details about OPSEC certified clusters, see the vendor documentation.

An important aspect of this is that packets sent from cluster members (as opposed to packets routed through the members) are hidden behind the cluster IP and MAC addresses. The cluster MAC is the:

- MAC of the active machine, in High Availability New mode.
- Multicast MAC, in Load Sharing Multicast mode.
- Pivot member MAC in Load Sharing Unicast mode.

This enables the members to communicate with the surrounding networks, but also has certain limitations, as described in Limitations of Cluster Addresses on Different Subnets (on page 99).
Configuration of Cluster Addresses on Different Subnets

There are two major steps required in order for ClusterXL to function correctly with cluster IPs on different subnets.

The first step is to create static routes on each cluster member, which determine the interface connected to the cluster’s network (the subnet to which the cluster IP belongs). Unless these entries are created, the OS cannot route packets to the cluster's network. No additional configuration is required for the cluster members. It is, however, important to note that the unique IPs given to the members must share common subnets on each “side” of the cluster (meaning, each interface on each machine must have an interface on every other machine using the same subnet).

The second step relates to the configuration of the cluster topology. Here the cluster IPs are determined, and associated with the interfaces of the cluster members (each member must have an interface responding to each cluster IP). Normally, cluster IPs are associated with an interface based on a common subnet. In this case these subnets are not the same. It must be explicitly specified which member subnet is associated with the cluster IP.

To specify the member network:
2. Click Edit Topology.
3. In the Edit Topology window, manually enter the IP address and subnet in the appropriate member interface fields.

Note that this interface actually refers to the cluster’s virtual IP address, as determined in the cluster topology.

Example of Cluster Addresses on Different Subnets

In this example, a single-gateway firewall separating network 172.16.6.0 (Side "A") from network 172.16.4.0 (Side "B") is to be replaced with a ClusterXL cluster. The cluster members, however, will use networks 192.168.1.0 for Side "A", 192.168.2.0 for Side "B" and 192.168.3.0 for the synchronization network (all network addresses given in this example are of class "C"). The addresses in italics are the cluster IP addresses. The resulting configuration is depicted below:

Figure 9-11 Cluster Addresses on Different Subnets

Configuring Static Routes on the Members

Each member should be configured with two static routes:

- One setting its 192.168.1.x IP address as the gateway for network 172.16.6.0
- One setting its 192.168.2.x IP address as the gateway for network 172.16.4.0.

To configure a static route on SecurePlatform, run sysconfig from the command prompt, choose Routing > Add New Network Route, and follow the instructions.
Configuring Cluster IP Addresses in SmartDashboard

To configure the cluster interface IP addresses:

1. In the Gateway cluster object Topology > Edit Topology window, edit a cluster interface, and open the Interface Properties window.
2. For each cluster interface, configure the Interface Properties window as follows:

<table>
<thead>
<tr>
<th>Table</th>
<th>Example ClusterXL Topology &gt; Interface Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cluster Interface A IP address</td>
</tr>
<tr>
<td></td>
<td>Cluster Interface B IP address</td>
</tr>
<tr>
<td>General tab</td>
<td>172.16.6.100</td>
</tr>
<tr>
<td>Member Networks tab</td>
<td>192.168.1.0</td>
</tr>
</tbody>
</table>

- All IP addresses have the Netmask 255.255.255.0

Note - Do not define Cluster IP addresses for the synchronization interfaces. The synchronization interfaces are also defined in the Edit Topology page of the Gateway Cluster object.

Limitations of Cluster Addresses on Different Subnets

This new feature does not yet support all the capabilities of ClusterXL. Some features require additional configuration to work properly, while others are not supported.

Connectivity Between Cluster Members

Since ARP requests issued by cluster members are hidden behind the cluster IP and MAC, requests sent by one cluster member to the other may be ignored by the destination machine. To allow cluster members to communicate with each other, a static ARP should be configured for each cluster member, stating the MAC addresses of all other machines in the cluster. IP packets sent between members are not altered, and therefore no changes should be made to the routing table.

Note - Static ARP is not required in order for the machines to work properly as a cluster, since the cluster synchronization protocol does not rely on ARP.

Load Sharing Multicast Mode with "Semi-Supporting" Hardware

Although not all types of network hardware work with multicast MAC addresses, some routers can pass such packets, even though they are unable to handle ARP replies containing a multicast MAC address. Where a router semi-supports Load sharing Multicast mode, it is possible to configure the cluster MAC as a static ARP entry in the router's internal tables, and thus allow it to communicate with the cluster.

When different subnets are used for the cluster IPs, static ARP entries containing the router's MAC need to be configured on each of the cluster members. This is done because this kind of router will not respond to ARP requests containing a multicast source MAC. These special procedures are not required when using routers that fully support multicast MAC addresses.
Manual Proxy ARP

When using static NAT, the cluster can be configured to automatically recognize the hosts hidden behind it, and issue ARP replies with the cluster MAC address, on their behalf. This process is known as Automatic Proxy ARP.

However, if you use different subnets for the cluster IP addresses, this mechanism will not work, and you must configure the proxy ARP manually. To do so, in SmartDashboard, select Policy menu > Global Properties > NAT Network Address Translation, and disable Automatic ARP Configuration. Then create a file called local.arp in the firewall's configuration directory ($FWDIR/conf).

Each entry in this file is a triplet, containing the:

- host address to be published
- MAC address that needs to be associated with the IP address
- unique IP of the interface that responds to the ARP request.

The MAC address that should be used is the cluster's multicast MAC defined on the responding interface, when using multicast LS, or this interface's unique IP, for all other modes.

For example, if host 172.16.4.3 is to be hidden using the address 172.16.6.25, and the cluster uses Load Sharing Multicast mode, add the following line to the local.arp file of Member 1:

172.16.6.25 00:01:5e:10:06:64 192.168.1.1

The second parameter in this line is the multicast MAC address of cluster IP 172.16.6.100, through which ARP requests for 172.16.6.25 will be received. On Member 2, this line will be:

172.16.6.25 00:01:5e:10:06:64 192.168.1.2

If the cluster is in unicast LS mode, or in HA mode, the entries on Member 1 and 2 will be:

172.16.6.25 00:A0:C9:E8:C7:7F 192.168.1.1

And

172.16.6.25 00:A0:C9:E8:CB:3D 192.168.1.2

where the second entry in each line is the unique MAC address of the matching local interface.

Connecting to the Cluster Members from the Cluster Network

Since the unique IPs may be chosen arbitrarily, there is no guarantee that these addresses are accessible from the subnet of the cluster IP. In order to access the members through their unique IPs, you must configure routes on the accessing machine, such that the cluster IP is the gateway for the subnet of the unique IPs. Following the above example, 172.16.6.100 should be the gateway for subnet 192.168.1.0.

Default Gateway on SecurePlatform

Run sysconfig > routing > add network route > add the routable network with its subnet, and choose the correct physical interface in this direction.

Now go to routing > add default gateway and add the IP address of the default (routable) gateway. This will usually be the IP address of the router in one of the cluster IP's subnet.

If you have the different subnets feature configured on more than one interface, repeat the addition of the network address (as above) for all these interfaces. (It is NOT required to define a default gateway for the other subnets as well.)

Anti-Spoofing

When the different subnets feature is defined on a non-external interface, the cluster IP in the Cluster Topology tab should not be defined with the Network defined by interface IP and Net Mask definition in the Topology tab of the Interface Properties window of the cluster interface. You must add a group of networks that contain both the routable network and the non-routable network, and define the Anti-spoofing for this interface as specific: network with this new group.
In the example shown in Figure 8-6 on page 209, suppose side "B" is the internal network, you must define a group which contains both 172.16.4.0 and 192.168.2.0 networks, and define the new group in the specific field of the Topology tab.

Moving from a Single Gateway to a ClusterXL Cluster

This procedure describes how to add a new gateway (Machine 'B') to a standalone Security Gateway (Machine 'A') to create a cluster.

As a prerequisite, there should be available IP addresses in a quantity equal to the number of new cluster members. If there are not, see: Configuring Cluster Addresses on Different Subnets.

On the Single Gateway Machine

If your single gateway installation uses the same machine for the Security Management server and the gateway:
1. Separate the Security Management server from the gateway, and place them on two machines.
2. Initialize SIC on the separated gateway (Machine 'A').

On Machine 'B'

1. Define an interface on machine 'B' for each proposed cluster interface and synchronization interface on machine 'A', with the same subnet. If the members exist on different subnets, see: Configuring Cluster Addresses on Different Subnets.
2. Install the Security Gateway on the machine. During the installation you must enable ClusterXL.

In SmartDashboard, for Machine 'B'

1. Create a ClusterXL object.
2. In the Cluster Members page, click Add, and select New Cluster Member.
3. Connect to machine 'B', and define its topology.
4. Define the Synchronization networks for the cluster.
5. Define the cluster topology. To avoid reconfiguring network devices, the cluster IP addresses should be the same as the addresses of machine 'A', on its proposed cluster interfaces.
6. Install the policy on the cluster, currently including member 'B' only.

On Machine 'A'

1. Disconnect all proposed cluster and Synchronization interfaces. New connections now open through the cluster, instead of through machine 'A'.
2. Change the addresses of these interfaces to some other unique IP address which is on the same subnet as machine B.
3. Connect each pair of interfaces of the same subnet using a dedicated network. Any hosts or gateways previously connected to the single gateway must now be connected to both machines, using a hub/switch.

Nota - It is possible to run synchronization across a WAN. For details, see Synchronizing Clusters over a Wide Area Network (see “Synchronizing Clusters on a Wide Area Network” on page 15).
In SmartDashboard for Machine 'A'
1. Update the topology of gateway A, either manually or by clicking Get Topology.
   If the IP address of the management interface was changed, the Get Topology action will fail. If this
   happens, manually change the main IP address in the gateway object and save the policy prior to
   performing an automatic topology fetch.
2. In the Cluster Members page, click Add, and select Add Gateway to Cluster.
3. Select machine 'A' in the window.
4. In the Edit Topology page, determine which interface is a cluster interface, and which is an internal or
   an external interface.
5. Install the policy on the cluster.

Adding Another Member to an Existing Cluster
1. On the cluster member, run cpconfig to enable ClusterXL.
2. Change the IP addresses of the new cluster member to reflect the correct topology (either shared IP
   addresses or unique IP addresses, depending on the clustering solution).
3. Ensure that all required Check Point products are installed on the new cluster member.
4. In the Cluster Members page of the Gateway Cluster object, either create a new cluster member (if it is
   a new Security Gateway machine) with the appropriate properties, or convert an existing Gateway to a
   cluster member.
5. If this is a new Security Gateway machine, ensure that SIC is initialized. In the Edit Topology page,
   ensure that the topology is correctly defined.
6. If the Cluster Mode is Load Sharing or New HA, ensure that the proper interfaces on the new cluster
   member are configured as Cluster Interfaces.
7. Install the security policy on the cluster.
8. The new member is now part of the cluster.

Configuring ISP Redundancy on a Cluster
If you have a ClusterXL Gateway cluster, connect each cluster member to both ISPs via a LAN using two
interfaces. The cluster-specific configuration is illustrated below.
Enabling Dynamic Routing Protocols in a Cluster Deployment

ClusterXL supports Dynamic Routing (Unicast and Multicast) protocols as an integral part of SecurePlatform. As the network infrastructure views the clustered gateway as a single logical entity, failure of a cluster member will be transparent to the network infrastructure and will not result in a ripple effect.

Components of the System

**Virtual IP Integration**

All cluster members use the cluster IP address(es).

**Routing Table Synchronization**

Routing information is synchronized among the cluster members using the Forwarding Information Base (FIB) Manager process. This is done to prevent traffic interruption in case of failover, and used for Load Sharing and High Availability modes. The FIB Manager is the responsible for the routing information.
The FIB Manager is registered as a critical device (Pnote), and if the slave goes out of sync, a Pnote will be issued, and the slave member will go down until the FIB Manager is synchronized.

**Failure Recovery**

Dynamic Routing on ClusterXL avoids creating a ripple effect upon failover by informing the neighboring routers that the router has exited a maintenance mode. The neighboring routers then reestablish their relationships to the cluster, without informing the other routers in the network. These restart protocols are widely adopted by all major networking vendors. The following table lists the RFC and drafts compliant with Check Point Dynamic Routing:

<table>
<thead>
<tr>
<th>Protocol</th>
<th>RFC or Draft</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSPF LLS</td>
<td>draft-ietf-ospf-lls-00</td>
</tr>
<tr>
<td>OSPF Graceful restart</td>
<td>RFC 3623</td>
</tr>
<tr>
<td>BGP Graceful restart</td>
<td>draft-ietf-idr-restart-08</td>
</tr>
</tbody>
</table>

**Dynamic Routing in ClusterXL**

The components listed above function "behind-the-scenes." When configuring Dynamic Routing on ClusterXL, the routing protocols automatically relate to the cluster as they would to a single device.

When configuring the routing protocols on each cluster member, each member is defined identically, and uses the cluster IP address(es) (not the member's physical IP address). In the case of OSPF, the router ID must be defined and identical on each cluster member. When configuring OSPF restart, you must define the restart type as signaled or graceful. For Cisco devices, use type signaled.

Use SecurePlatform's command line interface to configure each cluster member. **Figure 8-8** is an example of the proper syntax for cluster member A.

**Figure 8-8 Enabling OSPF on cluster member A**
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Launch the Dynamic Routing Module</td>
<td>[Expert@GWa]# router</td>
</tr>
<tr>
<td>localhost&gt;enable</td>
<td></td>
</tr>
<tr>
<td>localhost#configure terminal</td>
<td></td>
</tr>
<tr>
<td>Enable OSPF and provide an OSPF router ID</td>
<td>localhost(config)#router ospf 1</td>
</tr>
<tr>
<td>localhost(config-router-ospf)#router-id 192.168.116.10</td>
<td></td>
</tr>
<tr>
<td>localhost(config-router-ospf)#restart-type [graceful</td>
<td>signaled]</td>
</tr>
<tr>
<td>localhost(config-router-ospf)#redistribute kernel</td>
<td></td>
</tr>
<tr>
<td>Define interfaces/IP addresses on which OSPF runs</td>
<td>localhost(config-router-ospf)#network 1.1.10.10 0.0.0.0 area 0.0.0.0</td>
</tr>
<tr>
<td>and the area ID for the interface/IP address</td>
<td>localhost(config-router-ospf)#network 1.1.10.20 0.0.0.0 area 0.0.0.0</td>
</tr>
<tr>
<td>Exit the Dynamic Routing Module</td>
<td>localhost(config-router-ospf)#exit</td>
</tr>
<tr>
<td>localhost(config)#exit</td>
<td></td>
</tr>
<tr>
<td>Write configuration to disk</td>
<td>localhost#write memory</td>
</tr>
<tr>
<td>IU0 999 Configuration written to '/etc/gated.ami'</td>
<td></td>
</tr>
</tbody>
</table>

The same configuration needs to be applied to each cluster member.

As the FIB Manager uses TCP 2010 for routing information synchronization, the Security Policy must accept all traffic on port TCP 2010 between cluster members.

For detailed information regarding Dynamic Routing, see the Check Point Advanced Routing Suite guide.
Chapter 10

High Availability Legacy Mode

In This Chapter

Introduction to High Availability Legacy Mode 106
Example Legacy Mode Deployment 107
Planning Considerations 108
Configuring High Availability Legacy Mode 108
Moving from High Availability Legacy with Minimal Effort 113
Moving from High Availability Legacy with Minimal Downtime 114

Introduction to High Availability Legacy Mode

When configured to work in the High Availability Legacy Mode, all cluster members are assigned the same shared IP and MAC addresses. A shared interface is an interface whose MAC and IP addresses are identical to those of another interface.

The principal advantage of using this mode is that moving from a single gateway deployment to a High Availability cluster requires no changes to IP addresses or routing. Any switch or hub can connect to cluster interfaces. The disadvantage is that configuring this mode is complicated, and must be performed in a precise sequence. You must connect the Security Management server either to the cluster synchronization network or to a dedicated management network.
Example Legacy Mode Deployment

This example shows a typical High Availability Legacy mode deployment. The diagram relates the physical cluster topology to the required SmartDashboard configuration. It shows two cluster members: Member_A (the primary) and Member_B (the secondary) each with three interfaces. One for synchronization, one external shared interface, and one internal shared interface.

**Figure 10-13** Example Legacy HA Deployment

**Shared Interfaces IP and MAC Address Configuration**

With the High Availability Legacy mode, member interfaces connected to the same network (internal, external, DMZ, etc.) share common IP and MAC addresses. Each shared interface on cluster members connects to the appropriate network via a hub or switch.

Only one cluster member is active at any given time. Therefore, the outside world can see only the shared interfaces on one member at any given time.

In the above example deployment, the external interface (facing the Internet) uses the shared IP address 192.168.0.1 for both Member A and Member B. The internal interface (facing the local network) uses IP address 172.20.10.1 for both Member A and Member B.

**The Synchronization Interface**

State Synchronization between cluster members ensures that existing connections handled by the failed member are maintained during and after failover. The synchronization network passes connection synchronization and other state information between cluster members. Since this network carries sensitive security policy information, it is essential that this connection is secure. You can define more than one synchronization network for backup purposes.

To secure synchronization interfaces, they should be directly connected by a cross-cable, or in the case of three or more cluster members, by means of a dedicated hub, switch, or VLAN.

High Availability cluster members do not have to be synchronized. However, if they are not synchronized, active connections may be lost during failover.
Planning Considerations

IP Address Migration

Legacy mode deployments can be easily migrated from an existing stand-alone gateway configuration to a High Availability cluster. In such cases, we recommend that you assign the existing interfaces IP addresses on your stand-alone Security Gateway as the cluster addresses when feasible. Doing so will avoid altering current IPSec endpoint identities, and in many cases, will make it unnecessary to change Hide NAT configurations.

Security Management server Location

The Security Management server download Security Policies to all cluster members. This is only possible if the Security Management server can "see" them all at any given time. The Security Management server cannot be connected to any network that connects to member *shared interfaces*, because these interfaces are configured to use the same IP and MAC addresses.

The Security Management server must connect to the cluster synchronization network or a dedicated management network, because the members will have unique IP addresses.

Routing Configuration

Configure routing so that external network and internal networks are routable to each other.

For example, the sample deployment ("Example Legacy Mode Deployment" on page 107) shows routing configuration as follows:

- Internal networks are defined using **172.20.0.1** as the default gateway.
- The external router is configured with a static route such that network **172.20.0.1** is reached via **192.168.10.1**.

Switch (Layer 2 Forwarding) Considerations

The Cluster Control Protocol (CCP) makes use of layer two multicast. In keeping with multicast standards, this multicast address is used only as the destination, and is used in all CCP packets sent on "non-secured" interfaces.

A Layer 2 switch connected to non-secured interfaces, must be capable of forwarding multicast packets to switch ports, or within a VLAN, if it is a VLAN switch. It is acceptable that the switch forward such traffic to all ports, or to ports within the given VLAN. However, it is considered more efficient to forward to only those ports connecting cluster members.

Most switches support multicast by default. Please check your switch documentation for details.

If the connecting switch is incapable of forwarding multicast, CCP can be changed to use broadcast instead.

To toggle between these two modes:

- Use the command: `cphaconf set_ccp broadcast/multicast`

Configuring High Availability Legacy Mode

1. Obtain and install a ClusterXL central license on your Security Management server.
2. Disconnect the Security Gateways that are to become cluster members from switches and/or hubs.
3. Assign the same IP addresses to the shared interfaces on each member. To avoid network conflicts due to sharing MAC addresses, define the IP addresses before physically connecting members to the cluster topology.
4. Install and configure Check Point Security Gateway on all cluster members. Each member must use the identical version and build. Refer to the *R75.20 Installation and Upgrade Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=12269) for installation and initial configuration procedures. Do not reboot the members at this time.
5. Connect (or reconnect) the members to their switches and hubs. Make sure that you connect each interface to the appropriate physical network port. Connect each network (internal, external, Synchronization, DMZ, etc.) to a separate VLAN, switch or hub. No special switch configuration is required.

**Routing Configuration**

1. Configure routing so that communication with the networks on the internal side of the cluster is via the cluster IP address on the external side of the cluster. For example, in the sample deployment ("Example Legacy Mode Deployment" on page 107), the external router is configured as a static route such that network 10.255.255.100 is reached via 192.168.10.100.

2. Configure routing so that communication with the networks on the external side of the cluster is via the cluster IP address on the internal side of the cluster. For example, in the sample deployment ("Example Legacy Mode Deployment" on page 107), the internal router ports are configured with 10.255.255.100 as the default gateway.

3. Reboot the members. MAC address configuration will take place automatically.

**SmartDashboard Configuration**

1. In the Network Objects tree, right-click Check Point and then select Security Cluster.

2. In the Security Gateway Cluster Creation window, select Classic Mode.

3. In the Cluster Members page, click Add > New Cluster Member to add cluster members to the cluster. Cluster members exist solely inside the Gateway Cluster object. For each cluster member:
   a) In the Cluster Members Properties window General tab, define a Name and IP Address. Choose an IP address that is routable from the Security Management server so that the Security Policy installation will be successful. This can be an internal or an external address, or a dedicated management interface.
   b) Click Communication, and Initialize Secure Internal Communication (SIC).
   c) Define the NAT and VPN tabs, as required.
   d) You can also add an existing gateway as a cluster member by selecting Add > Add Gateway to Cluster in the Cluster Members page and selecting the gateway from the list in the Add Gateway to Cluster window.
   e) If you want to remove a gateway from the cluster, click Remove in the Cluster Members page and select Detach Member from Cluster or right-click on the cluster member in the Network Objects tree and select Detach from Cluster.

4. In the ClusterXL page,
   - Check High Availability Legacy Mode,
   - Choose whether to Use State Synchronization. This option is checked by default. If you clear this option, the cluster members will not be synchronized, and existing connections on the failed gateway will be closed when failover occurs.
   - Specify the action Upon Gateway Recovery (see What Happens When a Gateway Recovers? (on page 29) for additional information).
   - Define the Fail-over Tracking method.

5. In the Topology page, define the cluster member addresses. Do not define any virtual cluster interfaces. If converting from another cluster mode, the virtual cluster interface definitions are deleted. In the Edit Topology window:
   - Define the topology for each cluster member interface. To automatically read all the predefined settings on the member interfaces, click Get all members’ topology.
   - In the Network Objective column, define the purpose of the network by choosing one of the options from the drop-down list. Define the interfaces with shared IP addresses as belonging to a Monitored Private network, and define one (or more) interfaces of each cluster member as synchronization interface in a synchronization network (1st Sync/2nd Sync/3rd Sync). The options are explained in the Online Help. To define a new network, click Add Network.

6. Define the other pages in the Gateway Cluster object as required (NAT, VPN, Remote Access, etc.).
8. Reboot all the cluster members in order to activate the MAC address configuration on the cluster members.

**Configuring General Properties**

On the cluster **General Properties** page, configure the following:

1. Enter a unique name for this cluster object in the designated field.
2. Enter the main cluster IP address. This should be one of the cluster virtual interface IP addresses.
3. Select **ClusterXL** as a product installed on the cluster.
4. Enable **ClusterXL** and other Network Security Software Blades as appropriate.
Defining Cluster Members

On the Cluster Members page, click Add > New Cluster Member and configure the following:

1. In the Cluster Members Properties window General tab, enter a member Name and a physical IP Address, which must be routable from the Security Management server. This can be an internal or an external address, or a dedicated management interface.
2. Click Communication, and initialize Secure Internal Communication (SIC) trust.

Adding an Existing Gateway as a Member

To add an existing gateway as a cluster member, select Add > Add Gateway to Cluster on the Cluster Members page and select a gateway from the list in the Add Gateway to Cluster window.

Removing a Member

To remove a gateway from the cluster, click Remove in the Cluster Members page and select Detach Member from Cluster or right-click on the cluster member in the Network Objects tree and select Detach from Cluster.
Configuring ClusterXL Properties

To configure ClusterXL properties for the legacy mode:

1. Enable the High Availability option and select the Legacy mode.
2. Select the action to perform upon primary member recovery:
   - Maintain the current active cluster member
   - Switch back to a higher priority cluster member
3. For High Availability New deployments, State Synchronization is optional, but enabled by default. If you choose to disable State Synchronization, cluster members do not synchronize, and existing connections on the failed gateway will be terminated once failover occurs.
4. Select tracking options from the list.

Configuring the Cluster Topology

1. In the Topology page, click Edit Topology to define the virtual cluster IP addresses and at least one synchronization network.
2. In the **Edit Topology** window:

![Edit Topology window]

a) Define the topology for each member interface. To automatically read all predefined settings on member interfaces, click **Copy topology to cluster interfaces**.

b) In the **Network Objective** column, select an objective for each network from its list. The various options are explained in the online help. To define a new network, click **Add Network**.

3. Define the topology for each virtual cluster interface. In a virtual cluster interface cell, right click and select **Edit Interface**. The **Interface Properties** window opens.

a) In the **General** tab, **Name** the virtual interface, and define an **IP Address**.

b) In the **Topology** tab, define whether the interface is internal or external, and set up anti-spoofing.

c) In the **Member Networks** tab, define the member network and net mask if necessary. This advanced option is explained in Configuring Cluster Addresses on Different Subnets.

**Completing the Definition**

1. Define the other pages in the cluster object as required (**NAT, VPN, Remote Access**, and so on).


---

**Moving from High Availability Legacy with Minimal Effort**

This procedure describes how to move from High Availability Legacy mode to Load Sharing Multicast mode or to High Availability New mode, when the consideration is simplicity of configuration, rather than the minimal downtime.

The shared internal and external interfaces become cluster interfaces. The general IP address of the cluster therefore stays as an external cluster IP address.

**On the Gateways**

1. Run **cpstop** on all members (all network connectivity will be lost).

2. Reconfigure the IP addresses on all the cluster members, so that unique IP addresses are used instead of shared (duplicate) IP addresses.
Note - SecurePlatform only: These address changes delete any existing static routes. Copy them down for restoration in step 4.

3. Remove the shared MAC addresses by executing the command:
   \texttt{cphaconf uninstall_macs}
4. SecurePlatform cluster members only: Redefine the static routes deleted in step 2.
5. Reboot the members.

**From SmartDashboard**

In SmartDashboard, open the cluster object, select the \texttt{ClusterXL} tab, change the cluster mode from Legacy mode to new mode or to Load sharing mode. Then follow the Check Point Gateway Cluster Wizard. For manual configuration, proceed as follows:

1. In the \textbf{Topology} tab of the cluster object,
   - For each cluster member, get the interfaces which have changed since the IP addresses were changed. The interfaces which were previously shared interfaces should now be defined as Cluster interfaces.
   - Define the cluster IP addresses of the cluster. The cluster interfaces' names may be defined as you wish as they will be bound to physical interfaces according to the IP addresses.

   If the new IP addresses of the cluster members on a specific interface reside on a different subnet than the cluster IP address in this direction, the cluster members' network should be defined in the Members Network fields of the cluster interface (Configuring Cluster Addresses on Different Subnets).

2. Install the policy on the new cluster object (Security policy, QOS policy and so on).

**Moving from High Availability Legacy with Minimal Downtime**

This procedure describes how to move from Legacy Check Point High Availability to New Check Point High Availability or to Load Sharing while minimizing the downtime of the cluster.

The shared internal and external interfaces become the cluster interfaces. As the cluster members will need additional IP addresses these must be prepared in advance.

If downtime of the cluster during the change is not a major issue, it is recommended to use the easier process described in Moving from High Availability Legacy with Minimal Effort (on page 113).

Note - 1. Make sure that you have all the IP addresses needed before you start implementing the changes described here.
2. Backup your configuration before starting this procedure, because this procedure deletes and recreates the objects in SmartDashboard.

In this procedure we use the example of machines 'A' and 'B', with the starting point being that machine 'A' is active, and machine 'B' is on standby.

1. Disconnect machine 'B' from all interfaces except the interface connecting it to the Security Management server (the management interface).
2. Run \texttt{cphastop} on machine 'B'.
3. Change the IP addresses of machine 'B' (as required by the new configuration).

   Note - SecurePlatform only: These address changes delete any existing static routes. Copy them down for restoration in step 5.

4. Reset the MAC addresses on machine 'B' by executing \texttt{cphaconf uninstall_macs}. The Windows machine must be rebooted for the MAC address change to take effect.
5. SecurePlatform cluster members only: Redefine the static routes deleted in step 3.
6. In SmartDashboard, right-click member 'A' and select Detach from cluster.
7. In the **Topology** tab of the **Cluster Member Properties** window, define the topology of cluster member 'B' by clicking **Get...**. Make sure to mark the appropriate interfaces as **Cluster interfaces**.

8. In the **Cluster Object**, define the new topology of the cluster (define the cluster interfaces in the cluster's **Topology** tab).

9. In the **ClusterXL** page, change the cluster's **High Availability** mode from **Legacy Mode** to **New Mode** or select **Load Sharing** mode.

10. Verify that the other pages in the **Cluster Object** (NAT, VPN, Remote Access and so on) are correct. In Legacy Check Point High Availability, the definitions were per cluster member, while now they are on the cluster itself.

11. Install the policy on the cluster, which now only comprises cluster member 'B'.

12. Reconnect machine 'B' (which you disconnected in step 1) to the networks.

13. In this example the cluster comprises only two members, but if the cluster comprises more than two members, repeat steps 1-9 for each cluster member.

14. For **Load Sharing Multicast** mode, configure the routers as described in Table 4-5 on page 71.

15. Disconnect machine 'A' from all networks accept the management network. The cluster stops processing traffic.

16. Run **cphastop** on machine 'A'.

17. Run **cpstop** and then **cpstart** on machine 'B' (if there are more than two machines, run these commands on all machines except 'A').

18. Machine 'B' now becomes active and starts processing traffic.

19. Change the IP addresses of machine 'A' (as required by the new configuration).

20. Reset the MAC addresses of machine 'A' by executing **cphaconf uninstall_macs**. The Windows machine must be rebooted for the MAC address change to take effect.

21. In SmartDashboard, open the **Cluster Object** and select the **Cluster Members** page. Click Add > Add **Gateway to Cluster** and select member 'A' to re-attach it to the cluster.

22. Reconnect machine 'A' to the networks from which it was disconnected in step 15.

23. Install the security policy on the cluster.

24. Run **cpstop** and then **cpstart** on machine 'A'.

25. Redefine static routes

The cluster now operates in the new mode.
Chapter 11

Example cphaprob Script

The `clusterXL_monitor_process` script shown below has been designed to monitor the existence of given processes and cause failover if the processes die. It uses the normal pnote mechanism.

The `clusterXL_monitor_process` script is located in `$FWDIR/bin`.

```bash
#!/bin/sh
#
# This script monitors the existence of processes in the system. The process names should be written
# in the `$FWDIR/conf/cpha_proc_list` file one every line.
#
# USAGE :
# cpha_monitor_process X silent
# where X is the number of seconds between process probings.
# if silent is set to 1, no messages will appear on the console.
#
# We initially register a pnote for each of the monitored processes
# (process name must be up to 15 characters) in the problem notification mechanism.
# when we detect that a process is missing we report the pnote to be in "problem" state.
# when the process is up again - we report the pnote is OK.

if [ "$2" -le 1 ]
then
    silent=$2
else
    silent=0
```
fi
if [ -f $FWDIR/conf/cpha_proc_list ]
then
    procfile=$FWDIR/conf/cpha_proc_list
else
    echo "No process file in $FWDIR/conf/cpha_proc_list"
    exit 0
fi

arch=`uname -s`

for process in `cat $procfile`
do
    $FWDIR/bin/cphaprob -d $process -t 0 -s ok -p register > /dev/null 2>&1
done

while [ 1 ]
do

    result=1

    for process in `cat $procfile`
do
        ps -ef | grep $process | grep -v grep > /dev/null 2>&1

        status=$?

        if [ $status = 0 ]
        then
            if [ $silent = 0 ]
            then
                echo " $process is alive"
            fi
            #                      echo "3, $FWDIR/bin/cphaprob
            $FWDIR/bin/cphaprob -d $process -s ok report"
            else
                if [ $silent = 0 ]
                then
                    echo " $process is down"
                fi
            fi

    else
        $FWDIR/bin/cphaprob -d $process -s problem report
        result=0
fi

done

if [ $result = 0 ]

then
    if [ $silent = 0 ]
    then
        echo " One of the monitored processes is down!"
    fi
else
    if [ $silent = 0 ]
    then
        echo " All monitored processes are up "
    fi
fi

fi

if [ "$silent" = 0 ]
then
    echo "sleeping"
fi

sleep $1

done
Notes • 33

O
Old or too-new Arriving Updates • 66
On Machine 'A' • 101
On Machine 'B' • 101
On the Gateways • 113
On the Single Gateway Machine • 101
Operating System Compatibility • 33
Other Member Updates: • 66
Output of cphaprob [-reset] syncstat • 65
Overview • 46, 83

P
Performance Guidelines for Link Aggregation • 89
Planning Considerations • 108
Platform Specific Error Messages • 76
Pnote logs • 59
Preparing the Cluster Member Machines • 36, 43
Preparing the Switches and Configuring Routing • 42

Q
Queues: • 71

R
Reason Strings • 60
Receiving Queue Size • 71
Recommended Logging Options for High Availability • 51
Reconfiguring the Acknowledgment Timeout • 72
Recv Duplicate Retrans request • 68
Recv Retransmission requests • 67
Reducing the Number of Pending Packets • 95
Registering a Critical Device • 56
Registering Critical Devices Listed in a File • 57
Removing a Cluster Member • 50
Removing a Member • 39, 111
Removing IP Addresses from Slave Interfaces • 85
Reporting Critical Device Status to ClusterXL • 57
Routers • 32
Routing Configuration • 108, 109
Routing Switch • 32
Routing Table Synchronization • 103

S
Sample Configuration of PortFast Feature on a Cisco Switch • 92
SecureXL logs • 60
Security Management server Location • 108
Sending Queue Size • 71
Sent Retransmission Requests • 66
Setting Affinities • 89
Setting Critical Required Interfaces • 88
Setting Module Variables in IPSO 6.1 and Later • 93
Setting Slave Interfaces as Disconnected • 86
Shared Interfaces IP and MAC Address Configuration • 107
SmartDashboard Configuration • 109
SmartDashboard Configuration for OPSEC Clusters • 43
SmartView Monitor • 58
SmartView Tracker • 58
SmartView Tracker Active Mode Messages • 74
Starting and Stopping ClusterXL Using SmartView Monitor • 58
Starting the Cluster Member • 62
State logs • 59
Sticky Connections • 17
Stopping the Cluster Member • 61
Supported Logging Options for UTM-1 Clusters • 51
Switch (Layer 2 Forwarding) Considerations • 108
Switches • 32
Sync Related Error Messages • 74
Sync Statistics (IDs of F&A Peers - 1): • 65
Sync tick (ms) • 71
Synchronization Troubleshooting Options • 71
Synchronized Cluster Restrictions • 15
Synchronizing Clusters on a Wide Area Network • 15
Synchronizing Connection Information Across the Cluster • 11
Synchronizing Non-Sticky Connections • 14

T
TCP Out-of-State Error Messages • 75
The Check Point State Synchronization Solution • 11
The Cluster Control Protocol • 9
The clusterXL_monitor_process script • 116
The cphaconf Command • 61
The cphaprob Command • 52
The cphaprob Command in OPSEC Clusters • 45
The cphastart and cphastop Commands • 61
The cphastart and cphastop Commands in OPSEC Clusters • 45
The Need for Gateway Clusters • 8
The Sticky Decision Function • 17
The Synchronization Interface • 107
The Synchronization Network • 11, 24
Third-Party Gateways in Hub and Spoke Deployments • 18
Timed out Sync Connection • 67
Timers: • 71
Total Generated Updates • 67
Troubleshooting Bonded Interfaces • 91
Troubleshooting Synchronization • 64
Troubleshooting Workflow • 91

U
Unhold Pkt Events • 70
Unregistering a Critical Device • 57
Unsynced Missing Updates • 66
Upgrading to a UTM-1 Cluster • 50
Using the Wizard • 38
UTM-1 Clustering • 46

V
Verifying that a Cluster is Working Properly • 52
Verifying that the Bond is Functioning Properly • 86
Virtual IP Integration • 103
VLAN Support in ClusterXL • 80
VPN Tunnels with 3rd Party Peers and Load Sharing • 18

W

Warnings Regarding Use of PortFast • 92
What Happens When a Gateway Recovers? • 29
When Does a Failover Occur? • 29
Windows • 93
Workflow of Interface Bond in Load Sharing Mode • 87
Working with NAT and Clusters • 79
Working with OPSEC Certified Clustering Products • 42
Working with SmartView Tracker Active Mode • 95
Working with VLANs and Clusters • 80
Working with VPNs and Clusters • 78